

# UIUCnet

The University of Illinois Campus Network

Vol. 6 No. 1

Dec. 1992 - Jan. 1993

## Icon Key

### Audience



Novices



Experienced users



Network Administrators

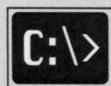


AISS clients



Everyone

### Platform/Operating System



PC compatibles (DOS™)



Macintosh™



X Window System™



NeXT™



CCSO'S IBM™  
mainframe, VMD



UNIX™ mainframes  
and workstations



Any platform

## Exploring the Power of the Internet Gopher

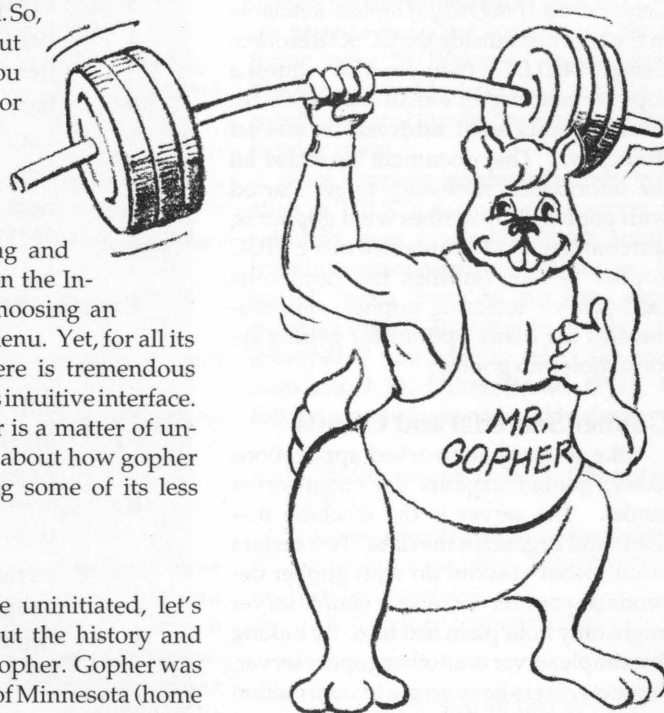
### INTERNET TREASURES



By now, most UIUCnet users have at least heard about *gopher*. The furry little rodent who burrows through *gopherspace* on the Internet has been featured twice in CCSO's *Updates* newsletter (vol. 3 no. 4 and vol. 3 no. 8) and once in the semi-monthly UIUC faculty/staff newspaper *Inside Illinois* (vol. 12 no. 11). National publications for computing and networking professionals and hobbyists (e.g., *MacWeek*, *Network World*, *Computer Shopper*) have also been tracking the development of this increasingly popular and ubiquitous Internet tool. So, why another article about *gopher*? Well first, if you haven't yet read about or seen *gopher*, you should make a point of it. *Gopher* is the only application that truly makes navigating and using many services on the Internet as natural as choosing an entree from a dinner menu. Yet, for all its elegant simplicity, there is tremendous power behind *gopher*'s intuitive interface. Unleashing this power is a matter of understanding a little bit about how *gopher* works and discovering some of its less obvious capabilities.

of the Golden Gophers) in an effort to provide the UM students and staff with a flexible Campus-Wide Information System (CWIS) for disseminating news, announcements, and other kinds of information to the university community. In order to make it easy for departmental information providers to maintain control over their own data, the *gopher* team sought to develop a "distributed document delivery system"—that is, a system in which the data could physically reside on multiple computers in multiple locations. Their solution was a TCP/IP-based client-server protocol and a set of applications that provided for the coordination and linking of

(continued on page 2)



### Back to Basics

For the sake of the uninitiated, let's review a little bit about the history and nature of the Internet *gopher*. *Gopher* was born at the University of Minnesota (home

## Gopher...

(continued from page 1)

multiple information servers across campus, while at the same time presenting that information to the end-user in a way such that it all appears to come from the same place.

Over the last two years, gopher has evolved from a system primarily intended to distribute text documents to a highly customizable environment for providing access to many different types of files and popular network services. Gopher has been adopted by hundreds of sites across the Internet—including the University of Illinois—as the CWIS/information server of choice. Today, the same method that was used to link multiple servers at the University of Minnesota campus is now used to link gopher servers all over the globe. The result is a seamless network of information servers, all of which can be easily accessed through a single, menu-driven interface.

Due to the superhuman efforts of co-administrators Paul Gibbs and Lynn Bilger, the University of Illinois gopher service is now internationally recognized as one of the best in gopherspace. Later on, we'll investigate some of its features in detail. But, if you haven't yet had a chance to access gopher or want to know how you can use it to distribute your own information, get a copy of the document called *Gopher at the University of Illinois*, available in the rack just outside the CCSO Resource Center, 1420 DCL (you can also request a copy by sending an e-mail message with your campus mail address to [uiucnet@uiuc.edu](mailto:uiucnet@uiuc.edu)). This document provides all the information necessary to get started with gopher. It describes what gopher is, summarizes what's contained in the UIUC gopher server, outlines the numerous methods for accessing gopher, and enumerates the many options for getting information into gopher.

## Gopher Server(s) and Clients

Like so many networked applications today, gopher exploits the client-server model. The server is the machine that holds and organizes the data. To a certain extent, what you can do with gopher depends on your server. A very simple server might only hold plain text files. By linking this simple server to another gopher server, however, users have access to information and services on both the simple server and

the server to which it is linked. Today most servers contain more than just text files and links to other servers. In addition to holding hundreds of text files, the main gopher server here at UIUC (a NeXT workstation) includes an engine for browsing and downloading files from popular ftp sites, gateways to the archie file archive database and the WAIS (Wide Area Information Servers) distributed-database system, direct links to several types of electronic phone books, the ability to do full-text searches on many of the documents archived on our local server and remote text databases, preconfigured telnet sessions for connecting to popular electronic library catalogs and information servers across the Internet, and, of course, links to every other gopher server in the world and all the unrestricted services they offer.

The Minnesota gopher development team is constantly working on expanding the capabilities of the server software. In order to maintain a state-of-the-art gopher server, the server administrator must keep

abreast of the latest server software releases and be willing to upgrade the server as necessary. Although the Unix-based server is the most powerful, it is also possible to set up a gopher server with limited capabilities on a Macintosh or PC. Such servers might be appropriate for a small department that wants to publish its own text-based information but does not have the resources to purchase and maintain a complex Unix workstation.

Now let's consider the client side of gopher. The client is the computer and software that communicates with the server. It provides a friendly front-end for the end-user to view and select the services available on the server and all its links. Gopher clients have been developed for many different types of computers and operating systems and often differ in terms of "look and feel" (see the related article on page 8, "How Six Gopher Clients Stack Up"). Nevertheless, all gopher clients have several features in common.

(continued on page 3)

## Gopher Object Types

### Normal Types:

- 0 Item is a file
- 1 Item is a directory
- 2 Item is a CSO (qi) phone-book server
- 3 Error
- 4 Item is a BinHexed Macintosh file
- 5 Item is DOS binary archive of some sort
- 6 Item is a UNIX uuencoded file
- 7 Item is an Index-Search server
- 8 Item points to a text-based telnet session
- 9 Item is a binary file
- T TN3270 connection

### Experimental Types:

- s Sound type. Data stream is a mulaw sound
- g GIF type
- M MIME type. Item contains *MIME* (Multipurpose Internet Mail Extensions) data
- h html type. (*HyperText Markup Language* used by the World Wide Web, a hypertext application for finding and accessing resources on the Internet)
- I Image type
- i "inline" text type (used by panda, a proprietary version of gopher used at the University of Iowa)



## Gopher...

(continued from page 2)

The most obvious similarity among gopher clients is that the information and services available on the server are presented to the end-user as a series of nested menus. This type of menu structure is intended to resemble a hierarchical file system, a concept already familiar to most computer users. When a user first connects to a server, he or she sees the top-level menu. This is more or less equivalent to the "root" directory of a tree-structured file system. Like a root directory, the top-level menu often contains files and other menus, which are analogous to subdirectories (or folders) in a file system. These submenus may, in turn, contain files, additional submenus, or other kinds of objects (such as telnet sessions, index searches, links to ph servers, etc.), and so on.

The gopher file system metaphor is more obvious in some clients than others. For example, menu items that contain submenus in the Unix *curses* client terminate with a slash (/), the standard symbol for a directory in the Unix environment. *TurboGopher* for the Mac and *NeXT Gopher 1.3* represent the menu hierarchy as folders within folders. Other clients identify menus in some consistent but less intuitive manner. All menus in the X Window System client, for instance, are preceded by a ">" symbol, and *PC Gopher II* for DOS uses the symbol <D>, which stands for directory.

Another characteristic shared by gopher clients is their ability to speak and interpret the gopher client-server protocol. While this might seem self-evident, it's important to note that as new options and services are added to gopher, new terms are added to the gopher vocabulary (in fact, a full-fledged extension to the gopher protocol called *Gopher+* has been proposed). An older client will not know what to do if a message from the server includes a term it doesn't understand. This won't necessarily result in something catastrophic. It just means that your client may not be able to make use of all of the services available through gopher. Additionally, even the most up-to-date client may have limited functionality due to the hardware constraints of the machine on which it is installed. For example, some gopher clients (*NeXT Gopher*, *Xgopher 1.2*, Unix *curses*, etc.) can actually play sound files.

The CMS client on VMD can display the names of sound files and assigns the label <sound> to them, but cannot play them. *PC Gopher II*, on the other hand, doesn't even know about sound files. It can neither display their names nor play them. Suffice it to say, not all gopher servers and clients are created equal.

## A Gopher Conversation

Before leaving the topic of clients and servers, it's worth taking a few moments to consider what actually takes place during a gopher session. When you start gopher, your client opens a TCP connection with a gopher server (usually the server at the address specified in the client's configuration file). The client sends a carriage-return/line-feed to the server, which in gophertalk means, "Tell me what you've got to offer." The server responds by returning a stream of carefully formatted text about the contents of the top-level menu, after which the TCP connection is closed. Yes, closed! Even though it appears as if your client is maintaining a continuous connection with the server, client-server conversations in gopher are typically very brief. The server returns just enough information to the client so that the client can initiate another TCP connection and perform another action such as retrieving a file or opening a menu. For example, when you connect to the U of I Gopher server with the Unix *curses* client you see the menu:

1. Welcome to the U of Illinois Gopher.
2. Campus Announcements (12/1/92)/
3. What's New? (12/3/92).
4. Information about Gopher/
5. Keyword Search of Gopher Menus <?>
6. U of Illinois Campus Information/
7. Champaign-Urbana & Regional Information/
8. Computer Documentation/
9. Libraries/
10. Newspapers, Newsletters, and Weather/
11. Other Gopher and Information Servers/
12. Phone Books (PH)/
13. Internet File Server (ftp) Sites/

The client actually receives much more information about each item in the menu than is displayed on the screen. For each menu choice, the server sends five separate pieces of information: 1) the object type, 2) the specific text that should be

displayed in the menu on the client, 3) a selector string for retrieving the object (usually the directory or path in which the object is located), 4) the domain name of the host on which the object resides, and 5) the telnet port number that listens for requests on that host. The raw information transmitted from server to client for item 1 in the menu above looks like this:

```
0Welcome to the U of Illinois Gopher
0/Welcome gopher.uiuc.edu 70
```

The first symbol in the text stream (shown in black ink) is the object type, in this case "0". Every item displayed in a gopher menu has an object type associated with it. It is the object type that tells the client what the specific item is. The client uses this information to determine how to display the item in the menu (for example, graphical clients typically display text documents as pieces of paper, menus as folders, ph searches as telephones, and so on) and what to do with the item, should the user decide to select it. Type 0 is a text file (see the table on page 2 for the list of object identifiers and the types they represent). Immediately following the object type is the actual text that is displayed in the menu (shown in blue ink). The next three pieces of information tell the client how to access item 1 on the menu. If you select menu item number 1, your client would open a TCP session with the host named **gopher.uiuc.edu** at port 70. Once connected, the client would send the selector string "0/Welcome" to the server. The server would respond by sending the complete text of the welcome message back to your client.

As it happens, **gopher.uiuc.edu** is the alias of UIUC's main gopher server, so selecting item 1 would initiate another brief interaction with our local server. But not all menu items on our local server point to objects that reside locally. For example, if you were to select the menu item called **12. Phone Books (PH)/**, the following information would be displayed:

1. U of Illinois at Urbana-Champaign <CSO>
2. Internet-wide e-mail address searches/
3. Phone books at other institutions/
4. WHOIS Searches/
5. X.500 Gateway (experimental)/

(continued on page 4)

## Gopher...

(continued from page 3)

If you then selected item number 2, the raw information sent back from the server would be:

1Internet-wide e-mail address searches /Phone Books/.other gopher.micro.umn.edu 70

Object type 1 means that the item is a menu (or directory). The domain name **gopher.micro.umn.edu** indicates that by selecting this item, you would not be opening a connection with our local server. Rather choosing this item will point your client to a menu on the server at the University of Minnesota. The menu appears as if it is on our local server, but in reality it comes from somewhere else. This is how gopher establishes transparent "links" with other servers.

Why should you care about any of this? I can think of several good reasons. First, there may be an occasion when you want to know where the information in gopher actually originates, something that is not necessarily apparent from the menu entry. Many gopher clients have the ability to display the raw information that lies behind a menu choice. With the Unix curses client, you can view this information by pressing the = (equal) sign. Second, occasionally a server will identify an object incorrectly. If you try to retrieve a binary file with your client, and the server has told your client that the file is plain text, the transfer will be unsuccessful. Having access to raw server data can help diagnose problems such as this. It's also fascinating to consider how effortlessly one can hop from one server to another in gopher without thinking about a single network address.

## Integrated Services

Gopher's strongest selling point is its ability to integrate a variety of network services into a single application, so that users don't have to learn multiple software packages, commands, and network addresses to take advantage of them—a revolutionary step towards making the Internet accessible to the common man. However, if you've ever had experience with commercially available integrated software—the kind that combines word processing, database, spreadsheet, graphics, and telecommunications capabilities into a single package—you may have ob-

served that the functionality and sophistication of the individual components are often sacrificed for the convenience of inter-application compatibility and ease of use. Is the same thing true for gopher's implementation of well-known Internet services such as ftp, telnet, archie, WAIS, etc.? Well, that's a difficult question to answer. For many services, the answer depends specifically on which gopher client you are using. Some services are "client intensive," and if the gopher client doesn't do its job well, it will pale when compared to a stand-alone counterpart. But there are at least a few instances where gopher's ability to pull together multiple resources actually makes it more powerful than using a stand-alone application. Let's take a closer look at several of the services offered through gopher and determine how they measure up to alternative methods of access.

## Gopher as Document Delivery System

From the outset, gopher was conceived as a document delivery system, and it's fair to say that this is one of the things that gopher does best. All gopher servers and clients, no matter how primitive, know how to handle text documents. Most clients can display text documents on the screen, save them to a file, and/or print them. A few clients also offer the option of mailing the document to another person on the Internet.

Here at UIUC, gopher's text delivery talents are fully exploited. There is hardly a local event, announcement, or news item that doesn't make it into gopher. **Special Campus Announcements** have their own top level entry on our gopher menu. One could spend hours, if not days, browsing through the menu called **U of Illinois Campus Information**. Lurking beneath this menu choice is literally everything you wanted to know about the U of I, but were afraid to ask, neatly organized into menus and submenus. The full text of several local publications including *The Daily Illini*, *Inside Illinois*, the *Campus Crime Bulletin*, and others can be found under the **Newspapers, Newsletters, and Weather** menu. Gopher is rapidly becoming the official vehicle for disseminating important UIUC text-based information.

## Gopher as FTP Client

Ftp, the TCP/IP file transfer protocol, is one of the areas in which gopher shines—that is, if you have the right client. On

UIUC's gopher server, the ftp option is listed on the main menu with the title **Internet File Server (ftp) Sites**. Several popular ftp sites are listed by name. Additionally, most of the well-known ftp sites in the world are organized alphabetically into submenus in this same menu. For most of the sites in the alphabetical listing, a brief summary of what the site contains is provided. The same method used to move up and down through gopher's menu hierarchy (usually point and shoot or point and click) can be used to browse the directory contents of any ftp site in the list. And, if you've got the right client, you can use the same technique to transfer any file from the remote ftp server to your client machine.

All gopher ftp transactions involve three computers: 1) the remote ftp host, 2) the gopher server providing access to that host, and 3) your client. Problems can occur anywhere along this pipe, but the most common problem is that many gopher clients can only display and transfer text files. So, when you are browsing an ftp site, it may look as if there are very few files available, when in fact, your client is only showing you the file types that it can actually transfer. For example, PC Gopher II cannot handle binary file transfers. When browsing ftp sites with this client, you will only see files with object type 0 (plain text). Then there are clients that will only show you the files it thinks you want to know about. The Unix curses client can transfer binary files, but it only displays files that it thinks will be of interest to a Unix user. Thus, files with the extensions **.zip**, **.exe**, **.sit**, etc. are not visible when browsing ftp sites with this Unix client. Finally, there are clients, such as the CMS software on VMD, that can display the names of all file types, but can only transfer ASCII files. This client is good for browsing ftp sites, but you have to exit gopher and run a separate ftp application to actually fetch the file. The only clients I know of that can successfully display and transfer all types of binary files are TurboGopher for the Mac and Xgopher 1.2. It's a good bet that most gopher clients will be able to display and transfer all file types as software development continues. But for now, gopher's ftp capabilities are definitely limited by the versatility of your client.

## Gopher as Archie Client

Gopher's implementation of the archie service exemplifies how the integration of

(continued on page 5)



## Gopher...

(continued from page 4)

applications can improve performance. The archie service, called **Search of Most FTP Sites (archie)**, can be found beneath the main menu item **Internet File Server (ftp) Sites**. Archie is a searchable database of the file holdings of all major anonymous ftp sites on the Internet. You can feed archie a filename (or part of a filename) and archie will return a list of all the ftp sites that have a file matching your query. Prior to gopher, the only way to query archie was to telnet to one of several archie servers or use a stand-alone archie client. Archie would send back the desired information. Then, if you wanted to actually get the file, you would have to open an ftp connection with one of the sites listed by archie, change to the specified directory, and execute the proper ftp commands.

The gopherarchie gateway is integrated with its ftp engine. Thus, performing an archie query on gopher will not only provide you with a list of ftp sites and directories, but can literally take you to one of those ftp sites. And, if your gopher client knows how to transfer the file, you can grab it during the same transaction.

Comparing an archie query using the Unix stand-alone client with the same query on gopher will demonstrate exactly how powerful gopher's archie service is. Suppose you wanted to locate the well know e-mail package called *Eudora*. Figure 1 on this page shows part of the results returned by doing an archie query on the character string "eudora" using the stand-alone archie client on one of CCSO's Unix mainframes. Once archie has listed all the ftp hosts on which a file or directory named "eudora" resides, the transaction is over.

The gopherarchie service can take you several steps further. Figure 2 shows the information returned by gopher on the same query. Every item followed by a double slash (/) is a directory. By selecting item 10 in the menu, you can view the entire contents of the **mac/eudora** directory on **ux1.cso.uiuc.edu** as shown in Figure 3. At this point, you have actually opened an ftp connection with ux1 and are inside the **mac/eudora** directory. From here, if your client permits, you can transfer any of the files listed in the directory to your local computer. Unfortunately, because ftp and archie are tied together on gopher, many of the same limitations that apply to ftp also apply to archie.

```
Host ux1.cso.uiuc.edu

Location: /mac
  DIRECTORY drwxr-xr-x      512 Nov 18 00:05 eudora
Location: /mac/eudora
  FILE -rw-r--r--      459636 Oct 18 1991 eudora1.2.2.i.si
```

Figure 1: Partial results of an archie search on the character string "eudora" using the stand-alone Unix archie client.

```
Internet Gopher Information Client v1.01

Search of Most FTP sites (archie): eudora

1. world.std.com/src/unixmac/eudora//
2. metten.fenk.wau.nl/pub/mac/eudora//
3. mcsun.eu.net/comp/mac/mail/eudora//
4. akiu.gw.tohoku.ac.jp/pub/mac/comm/MacTCP/eudora//
5. mailer.cc.fsu.edu/pub/mac/eudora//
6. ftp.uu.net/systems/mac/eudora//
7. pinus.slu.se/pub/mac/mail/eudora//
8. dorm.rutgers.edu/pub/Mac/eudora//
9. src.doc.ic.ac.uk/computing/systems/mac/eudora//
--> 10. ux1.cso.uiuc.edu/mac/eudora//
11. miki.cs.titech.ac.jp/pub/Mac/eudora//
12. biom3.univ-lyon1.fr/pub/mac/comm/eudora-122.hqx <HQX>
13. ..pub/nfs-mounted/biom3.univ-lyon1.fr/mac/comm/eudora-122.hqx <HQX>
14. luga.latrobe.edu.au/pub/network/eudora//
15. ..ftpmail/solaris.ims.ac.jp/pub/unix/mac/eudora//
16. ..solaris.ims.ac.jp/pub/unix/mac/eudora-J/eudora1.2.2J6.sit.hqx <HQX>
17. ..solaris.ims.ac.jp/pub/unix/mac/eudora-J/eudora1.2.2J64.sit.hqx <HQX>
18. plaza.aarnet.edu.au/micros/mac/mailers/eudora//

Press ? for Help, Q to Quit, U to go up a menu      Page: 1/6
```

Figure 2: Partial results of an archie search on the character string "eudora" using the gopherarchie service.

```
Internet Gopher Information Client v1.01

ux1.cso.uiuc.edu/mac/eudora/

--> 1. .notar.
2. 1.2Changes.
3. 1.2man.glue.sit.hqx <HQX>
4. 1.2man.pm.sit.hqx <HQX>
5. 1.3Changes.
6. README.
7. ReleaseNotes.
8. appendix-d.
9. beta/
10. comm1.i.sit.hqx <HQX>
11. diskcopy.hqx <HQX>
12. doc.i.sit.hqx <HQX>
13. eudora1.2.2.i.sit.hqx <HQX>
14. intl/
15. oldbeta/
16. serialpop.c.
17. tables/

Press ? for Help, Q to Quit, U to go up a menu      Page: 1/1
```

Figure 3: The file contents of menu item 10 in Figure 2 above. Items that are files in this menu can be ftp'd to the client machine. Items that are directories (followed by a slash) can be opened just like any other gopher menu.

## Gopher as Telnet and Tn3270 Tool

Gopher has the ability to launch pre-configured telnet sessions. Many gopher servers, including our own, provide access to on-line library catalogs and other kinds of information servers. With gopher, you can easily access these systems without knowing their domain name or IP address by simply making a menu choice. However, when you select a telnet session

from a gopher menu, it is important to realize that your gopher client is not actually opening the telnet session. In fact, choosing a telnet session will cause you to temporarily leave gopher. The gopher client will look for the telnet application on your client machine and pass the necessary address information off to the telnet program. Gopher will also display any

(continued on page 6)

## Gopher...

(continued from page 5)

special information you need to know, such as the login id and password to give the remote host. Your telnet program will then try to contact the specified host using the information provided by gopher. When you close the session, you will be returned to gopher.

Whether a telnet session is successful or not depends on several factors. Most critical is making sure that your gopher client knows how to find your telnet client. The gopher clients on CCSO machines have been configured to launch telnet sessions for you. If you install a gopher client on your desktop computer, you may have to do something special so that gopher knows how to find telnet.

Gopher can also launch remote login sessions with computers that require 3270 terminal emulation, such as IBM mainframes. In principle, tn3270 sessions work just like telnet sessions in gopher and are subject to the same restrictions. Thus, gopher must be able to find the tn3270 program on your client machine. Additionally, your gopher client must know about the tn3270 object type. Neither PC Gopher II nor the CMS Gopher on VMD recognize the tn3270 object type. The DOS client simply does not display tn3270 sessions in its menu system. The CMS client displays the session as type T but returns the message "cannot process this type."

In general, the ability to open telnet and tn3270 sessions with gopher is a tremendous convenience. The gopher server at the University of North Texas, which you can find under **Other Gopher and Information Servers/Recommended Gopher Servers for Exploration/University of North Texas/**, has an exhaustive menu of telnet and tn3270 sessions to reach library catalogs all over the world. Although it is not possible to open a remote login session with a host that is not listed in a gopher menu, if you would like certain libraries or information servers added to the list of libraries and terminal-based sessions on the UIUC gopher server, simply send an e-mail request to our local gopher administrator at the address [gopher@uiuc.edu](mailto:gopher@uiuc.edu).

## Gopher as WAIS Client

Among the many services offered by gopher is a gateway to the WAIS (Wide-Area Information Servers) databases. WAIS is a collection of distributed data-

bases, which can be searched by keyword and cover a wide variety of subjects. (A detailed description of WAIS and how it works is given in the October 1992 issue of *UIUCnet*, vol. 5 no 6.) The gopher implementation of WAIS lacks many of the features found in dedicated WAIS clients. Most notably, with gopher you can only search one WAIS database at a time, and WAIS's unique search refining tool called *relevance feedback* is not available. However, gopher does offer one feature that most WAIS clients do not: it allows you to see the names and search all of the public WAIS databases that are available. Perhaps someday, a full-featured WAIS client will be built in to the gopher software.

## Gopher as Electronic Phone Book

Under the menu item **Phone Books (PH)** is a collection of electronic phone directories (often called *white pages*) that can be used to look up the e-mail address and/or other information about people on the Internet. The first item in this menu, **U of Illinois at Urbana-Champaign**, provides access to our local CCSO Nameserver, also known as ph. Many other institutions have used the ph program to set up their own electronic directory services. They can be found mixed in with other types of electronic directories in the submenu **Phonebooks at other institutions**.

Each gopher client represents ph directories in some consistent way. Some clients use the term **CSO**, which stands for CCSO Nameserver. Others use an icon resembling a telephone or telephone book. In any case, ph searches comprise a specific object type in gopher and are handled somewhat uniquely. Most searches conducted in gopher are actually handled by a gopher server—that is, the client asks you to enter a search string, it then sends your search string to a gopher server, the server looks for items matching your search string, and finally the results of the search are returned to your client. Ph queries, on the other hand, involve a direct conversation between the gopher client and a ph server; a gopher server does not participate in these transactions (except to provide the address of the ph server). In order for a gopher client to conduct a ph search, it must know how to speak the ph protocol, which is something quite different than the gopher protocol. Most gopher clients today have a built-in ph client. Some are easy to use but not very powerful, others are powerful but not very intuitive,

and a few simply don't work. Also, unlike many stand-alone clients, the ph clients built in to gopher do not allow a user to log in to a Nameserver to modify his or her own entry. Despite these minor shortcomings, between the many ph Nameservers and other electronic directory services (such as **whois** databases, the experimental **X.500** directory, and the utility called **netfind**), gopher offers a set of comprehensive tools for finding someone on the Internet, unmatched by any other client-server application.

## Gopher as File Viewer/Player

If you are fortunate enough to have the right software and hardware, gopher can be used to view images and play sounds. Sounds and images are experimental object types in Gopher and only a few clients know what to do with them. Moreover, like the telnet function in gopher, sound and image files are not actually played or viewed by the gopher client. The client looks for another program on the client machine that knows how to process the file. For example, if the application *Giffer* is installed on your Mac, when you download a *GIF* (graphics interchange format) file, TurboGopher will ask you if you would like Giffer to display it for you. Similarly, Xgopher 1.2 will pass GIF files off to the X application called *xloadimage*.

## Finding Things in Gopher

Well, after many words and pages, we've only begun to scratch the surface of gopher, which leads us to the last major topic in this article—with so much information on so many gopher servers all over the world, how does one actually find specific information in gopher, and once found, how can one keep track of where the information is located? Gopher actually provides several tools for locating information. There is a special object type in gopher called an *index-search server*. Index searches often have the word "search" as part of their menu entry and, like other object types in gopher, have a distinctive abbreviation or icon associated with them. A gopher administrator can create an index search for any large body of text contained in a gopher menu. This can help you to rapidly zero in on the documents of interest. For example, if you are searching the electronic version of the *Daily Illini* for all recent articles about the women's volleyball team, you could go to the **Daily**

(continued on page 7)



## Gopher...

(continued from page 6)

Illini Newspaper menu and choose the **Word Search of Latest Month** item. When asked to enter a search string, enter the word "volleyball." All DI articles from the last month that contain the word "volleyball" will be listed as a separate menu.

But how did we find the *Daily Illini* in the first place? The UIUC main gopher server has a wonderfully useful resource on the top level menu called **Keyword Search of Gopher Menus**. This item contains an index of the titles of every menu on our server. To find the *Daily Illini* amidst the megabytes of information and multitudes of menus on the server, do a **Keyword Search of Gopher Menus** on the string "daily illini." Or, suppose you know that there is a link to a searchable version of *Roget's Thesaurus* somewhere on our server, but you haven't a clue where to look. Just do a keyword search on the word "thesaurus" and you will be taken directly to the **Roget's Thesaurus** menu.

**VERONICA.** The **Keyword Search of Gopher Menus** handles searching menus on our own gopher server, but what about the hundreds of other servers on the Internet? Let me introduce you to **VERONICA**. **VERONICA** stands for "very easy rodent-oriented net-wide index to computerized archives" and does for gopher whatarchie does for anonymous ftp archives (now we're all waiting for Betty, Reggie, and Jughead to appear on the Internet). **VERONICA** is a utility that indexes the titles of all levels of menus for most gopher sites on the Internet. **VERONICA** works like any other index search. You enter a word or group of words that you are looking for, and **VERONICA** creates a custom menu of all titles on all menus throughout gopherspace that match your query. By selecting an item in the custom menu, you are transparently connected to the gopher server on which it is located.

**VERONICA** is a relatively new service and there are still a few problems to be worked out. First of all, **VERONICA** is very slow and bound to become slower as more users begin to use it. Developers are hoping to eventually distribute **VERONICA** searches among multiple servers, which should lessen the burden of any single **VERONICA** server. Secondly, many gopher servers point to popular items on other gopher servers leading to a great deal of redundancy among servers.

**VERONICA** does not eliminate duplicate items, so a given search can result in many repetitions of the same service, document, or menu. Finally, **VERONICA** does not tell you where each title comes from. If your client has the ability to show the technical information behind each menu item (as described earlier in this article), then you can find out the domain name of the source, but not all clients have this capability and not all users know how to interpret this information. **VERONICA** can be found on our local server by doing a **Keyword Search of Gopher Menus** on the string "veronica" or by browsing the menu called **Other Gopher and Information Servers**.

**Bookmarks.** Navigating gopher is normally an up or down proposition. You can gradually work your way down through a series of menus and wend your way back up again, but lateral movement is generally not possible. Suppose you find a file or service in gopher that you know you'll want to access time and time again in a menu far removed from the top. Most gopher clients allow you to create and save so-called *bookmarks*. A bookmark keeps track of exactly where a gopher item is located, regardless of whether it resides on your local server or a remote gopher server. To go to that item again, you need only ask gopher to display your bookmarks and select the bookmark as you would any other gopher item. Creating a collection of bookmarks is like creating your own customized gopher menu. The commands for creating, viewing, and selecting bookmarks vary from client to client. In Turbo Gopher for the Mac, for

instance, bookmarks are created by selecting a gopher menu item and then choosing the **Set Bookmark...** option from the Gopher menu (see Figure 4 on this page). Consult the on-line help or written documentation of your gopher client for more details on bookmarks.

## Gopher Broke?

As we've seen, a gopher session can involve many computers and many applications. It's not uncommon, however, for things to go wrong in gopher. Occasionally a specific server function like ftp will go down, or a server can go down altogether. And, of course, certain clients simply cannot perform specific tasks. As you become well acquainted with your gopher client, you'll get a sense of what it can and cannot do. If you perform an operation regularly and suddenly it doesn't work, the problem probably lies with the server. If on the other hand, you've never had success doing something like a tn3270 session with your client, chances are that it does not support this function or is improperly configured. Hopefully the foregoing explanation of what takes place during various types of gopher transactions will help you to determine the source of any difficulties you might encounter. If you suspect that there is a problem with our local server, send an e-mail message to [gopher@uiuc.edu](mailto:gopher@uiuc.edu) (you can also send comments and suggestions about gopher to this address).

Despite occasional problems, gopher is likely to open up the Internet to a much wider audience. Whether you are a power user or network neophyte, this rapidly evolving network tool has much to offer. In its infancy, gopher was called a distributed document delivery system. Today it is referred to as a distributed information delivery system. Who knows what tomorrow may bring? ■

-Lynn Ward

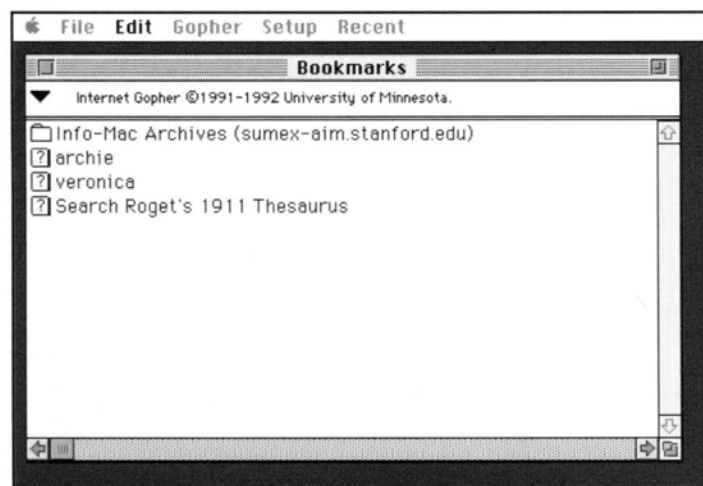


Figure 4: A menu of bookmarks saved in TurboGopher for the Macintosh. Each menu item in the Bookmarks window can be instantly accessed by selecting it.

# How Six Gopher Clients Stack Up

## CLIENTS FOR CLIENTS



The cover story of this issue describes how the Internet gopher can be a very powerful tool if you're using the right client software and if that client is properly configured. Over the last several months, I've had the opportunity to work with about ten different gopher clients, several of which have already been superseded by new, more powerful applications. To describe each of these clients in detail would fill a book, and for most gopher clients there is a user's guide, on-line help, and/or a man page that gives technical information and instructions for use. Below is a brief sketch of six gopher clients, each for a different computing platform. This review highlights the strengths and weaknesses of each client and, in a few cases, provides some important undocumented technical information. A summary of the features of each client can be found in Table 1 on page 10. All of the clients discussed in this review can be downloaded from the anonymous ftp host [boombox.micro.umn.edu](http://boombox.micro.umn.edu). They are located in subdirectories (named according to platform) under the `/pub/gopher` directory. Several clients are also available at the CCSO Resource Center, 1420 DCL.

**Note:** The comments below refer to the specific version of the client that was reviewed. Since many developers continue to work at improving their software, some of the bugs and other peculiarities described here may have been or will be fixed in a later release.

### Xgopher 1.2—the Latest and Greatest

One of the newest kids on the block is the latest release of *Xgopher*, an X Window System client written by Allan Tuchman, senior research programmer here at CCSO. *Xgopher* can handle just about any gopher function that's currently available. The interface is very simple to use (see Figure 1 on page 9). Most commands are available by clicking on buttons in the main window. Less commonly used commands are tucked away in a pull-down menu called **Other Commands**.

*Xgopher* provides support for a variety of different file or *object* types including: image files (image files are displayed by *xloadimage*, a separate utility that can display many different types of graphics file formats), sound files (sound files can be played if *Xgopher* is installed on a workstation that supports sound), Unix binary files, ph queries, index searches, and telnet and tn3270 sessions. An options panel under the **Other Commands** menu allows you to tell *Xgopher* whether you want to see all files or just the file types that it knows about. If you opt to see all files, *Xgopher* will display unknown types with a `<???` prefix. Any file type in a gopher menu or directory, even unknown types, can be transferred to the client machine (for unknown file types, *Xgopher* defaults to binary transfer mode) with the **copy selected item to file** command.

The bookmarks option is also extremely flexible in *Xgopher* 1.2. *Xgopher* can maintain multiple bookmark files, any of which may be loaded during a single session. Also, the bookmark file is compatible with the Unix curses client, so you can use the same bookmarks when dialing in from home.

The ph client in *Xgopher* is fairly powerful, but not quite as intuitive as other implementations. The user is prompted to enter a query in the **Query name** field and results are displayed in the box below. A **Show Fields** menu displays the names of the fields available on the selected ph server and the intended contents of those fields. There is some minimal on-line help, but in order to do anything more complex than a simple query on the **name** field, one must already be well acquainted with the syntax of ph commands.

X Window System gurus can totally customize *Xgopher* by modifying the application re-

sources file called *Xgopher*. This plain text file contains all the settings for *Xgopher*, including the host name of the default gopher server; the name of the default bookmarks file; the prefixes used for each gopher object type; the default screen colors; the default fonts; the specific text used for all menus, buttons, and dialog boxes; the specific applications used to play sound files, display image files, and conduct telnet and tn3270 sessions, and many other parameters. Any of the default settings can be changed to suit the preferences of the end-user.

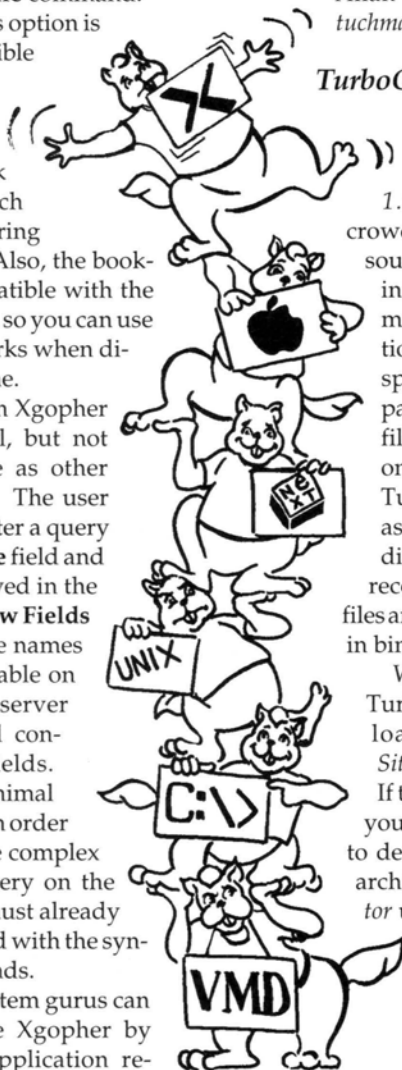
*Xgopher* is installed on `ux1`, `ux2`, `ux4`, `uxa`, and `uxh` and can be accessed by anyone with an X terminal or a desktop computer running an X server application such as *MacX*. Questions and comments about *Xgopher* should be directed via e-mail to Allan Tuchman at the address [a-tuchman@uiuc.edu](mailto:a-tuchman@uiuc.edu).

### TurboGopher 1.0.5 for the Mac

Although there are several gopher clients available for the Macintosh, *TurboGopher* 1.0.5 stands out among the crowd. *TurboGopher* cannot play sound files or display raw gopher information, but it handles almost every other gopher operation with the greatest of ease and speed. The ftp function works particularly well. To transfer a file to your Mac, just double click on its icon. If the file is BinHexed, *TurboGopher* will un-BinHex it as it transfers the file to your local disk. This gopher client even recognizes Unix and DOS binary files and will intuitively transfer them in binary mode.

When you get your copy of *TurboGopher*, be sure to download the helper-applications *SitExpand*, *CptExpand*, and *Giffer*. If these programs are installed on your Mac, *TurboGopher* will offer to decompress files that have been archived with the *StuffIt* or *Compactor* utilities (files with a `.sit` or `.cpt` extension) and display GIF image files immediately after such files have been downloaded.

(continued on page 9)





## Gopher Clients...

(continued from page 8)

The ph client in TurboGopher is very easy to use. Unfortunately it has a major bug. You can only enter a single name when doing a search on the **name** field. If you enter a first and a last name, the client sends the query to the server improperly and you'll get the message "501: No matches to your query." Since the **name** field is the most common field for doing ph lookups, this is a big problem. In order to do a query with TurboGopher's ph client, the person you are looking for must either have a very unique last name or you must know some additional information about the person, such as his or her phone number or address.

TurboGopher supports both telnet and tn3270 sessions only if the appropriate applications (NCSA Telnet and Brown's TN3270) have been installed on your hard disk. Additionally, in order to conduct a tn3270 session, a copy of your **config.tel** file must be located in your System Folder.

TurboGopher is installed on all of the Macs at CCSO's computing sites and is also available at the CCSO Resource Center. The software is copyrighted by the University of Minnesota, but can be copied and distributed freely.

### NeXT Gopher 1.3

For those fortunate enough to have a NeXT workstation sitting on their desk, *NeXT Gopher 1.3* offers a slick interface and good overall functionality. Two noticeable shortcomings are its inability to successfully display and transfer files other than plain text (supposedly, unsupported object types can be displayed by typing a special command at the terminal prompt, but this doesn't seem to work) and the absence of a **bookmarks** option. With the help of supporting applications, sound files can be played with exquisite clarity and GIF files can be viewed. Gopher files can also be mailed to other users with the NeXT client. One curious aspect of NeXT Gopher is that it does not have a built-in ph client. Instead it relies on a separate program, *NeXT Ph*, which comes bundled with the distribution. All in all, NeXT Gopher is probably the "prettiest" client around, but it still has some catching up to do in order to match the X and Mac clients.

### The Unix Curses Client 1.01

The Unix curses gopher client was de-

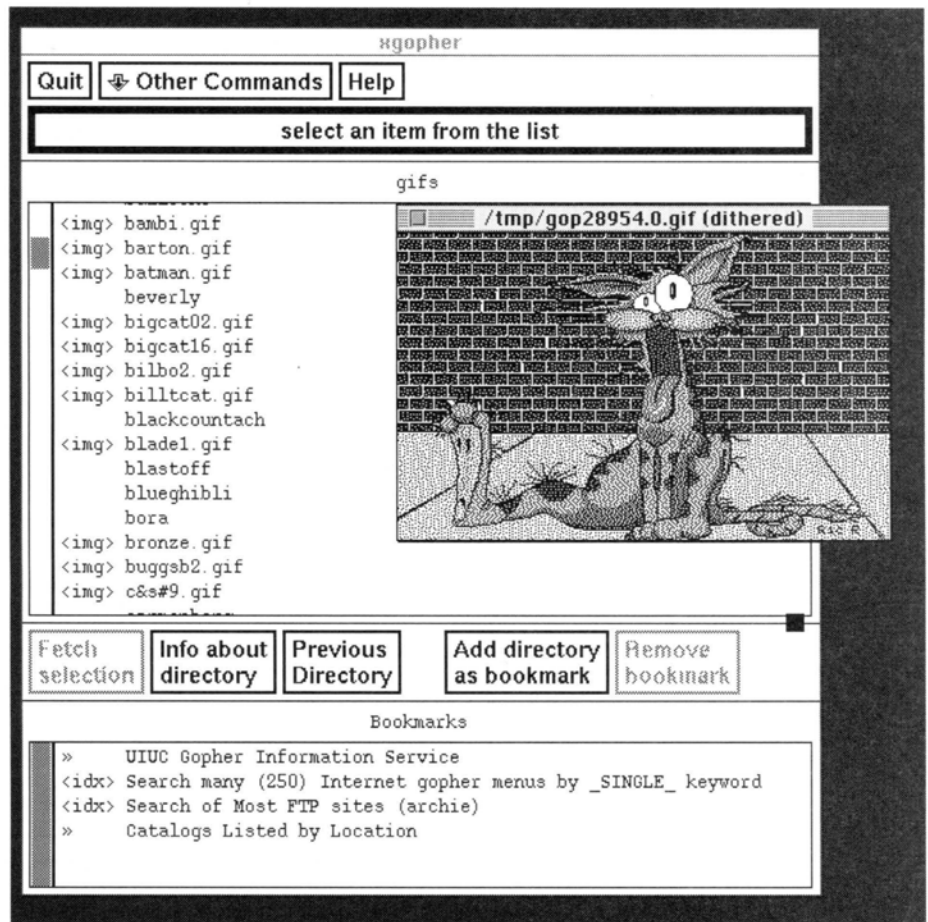


Figure 1: Xgopher 1.2 for the X Window System. An image file (Bill the Cat) is displayed in the foreground and the main Xgopher window is in the background.

signed to run on dumb terminals, so there are no fancy icons or mouse support. Navigating menus involves either moving the cursor to the desired item or entering its number and then pressing the **<return>** key. But don't let this simple interface mislead you. The curses client is fairly intelligent, although sometimes uncooperative. When installed on a Unix workstation, this gopher client can display image files and play sound files. Remote users with dumb terminals or terminal emulators can use the curses client to conduct archie searches, telnet and tn3270 sessions, ph queries, and index searches. Unfortunately, although the ftp client is capable of transferring any type of binary file, it only displays the names of ASCII files, BinHexed files, and Unix binaries. This can be very misleading when browsing ftp sites. Strangely enough, other types of files such as DOS executable and archived files will be displayed if you do an archie search, and these files can be successfully ftp'd once they are displayed.

Version 1.01 of the curses client doesn't do a very good job at recognizing object types. When it displays executable DOS files during an archie search, it assigns them an **<HQX>** type, which stands for BinHex. Tn3270 sessions work, but there is no type identifier next to them.

If you are new to the Unix curses client, be sure to check out the on-line help with the **? command**. It will tell you how to display technical information on an item, navigate menus, add bookmarks, and customize your gopher environment. You can change the program defaults of the curses client by executing the **O (options) command**. Within the options menu, you can specify the default pager, print command, mail command, etc. Bookmarks and your program defaults are saved in a **.gopherrc** file located in your home directory.

The curses gopher client installed on CCSO Unix machines is supported by CCSO staff. If you have a question or

(continued on page 10)

## Gopher Clients...

(continued from page 9)

problem, send an e-mail message to [gopher@uiuc.edu](mailto:gopher@uiuc.edu) or call the CCSO systems consultants at 333-6133.

### PC Gopher II 1.05r3

While there are several Gopher clients available for the DOS environment, PC Gopher II is the only one that does not require the user to load a commercial TCP/IP stack such as FTP Software's *PC/TCP*

kernel or Novell's *LAN Workplace for DOS*. This client uses the extended IBM character set to create a pseudo-graphical interface, complete with pull-down menus, sizable windows with scroll bars, and support for a Microsoft-compatible mouse. Let's first consider the strong points of the DOS client. PC Gopher II supports index searches, has a respectable and fully functional ph client, can display the technical information about an item, and supports bookmarks. Unfortunately, the bookmarks feature is limited to selecting menus only,

not items within menus such as an index search.

PC Gopher II also supports telnet sessions if it is properly configured. Be sure the **Allow Telnet Sessions** box is checked in the configuration window. Also, in order to conduct a telnet session, you must set a DOS environment variable so that gopher knows where to find your telnet application. To accomplish this, a line can be added to your **autoexec.bat** file or the batch file that starts gopher.

(continued on page 16)

Supported Features	Xgopher 1.2	TurboGopher 1.0.5	NeXT Gopher 1.3	Unix Curses Gopher 1.01	PC Gopher II 1.05r3	CMS Gopher 2.2.0
displays text files	yes	yes	yes	yes	yes (size limited by available memory)	yes
displays image files	yes	yes (GIFS only)	yes	yes (on certain Unix workstations)	no	no
displays names of all menu items/files regardless of object type	yes (the user can opt to display all filenames, even unsupported types)	yes	yes (according to documentation, but no success during testing)	no	no	yes
plays sound files	yes	no	yes	yes (on certain Unix workstations)	no	no
transfers text files	yes	yes	yes	yes	yes	yes
transfers binary files (limited types)	yes (unknown types are transferred in binary mode by default)	yes	no	yes	no	no
transfers all binaries	yes	yes	no	no	no	no
archie searches on all file types	yes	yes	no	yes	yes (sometimes)	yes
archie searches on text files	yes	yes	yes	yes	yes (sometimes)	yes
WAIS searches	yes	yes	yes	yes	yes	no
telnet sessions	yes	yes	yes	yes	yes	yes
tn3270 sessions	yes	yes (config.tel file must be in System folder)	yes	yes	no	no
ph searches	yes (interface not very intuitive)	yes (a bug limits searches on the "name" field to a single name)	yes (uses separate ph client, NeXT Ph)	yes	yes	yes
whois searches	yes	yes	yes	yes	yes	no
index-searches (general)	yes	yes	yes	yes	yes	some, but not all
displays technical info about menu item	yes	no	yes	yes	yes	no
recognizes most object types	many, but not all	yes	no	no	no	yes
bookmarks	yes (supports multiple bookmark files)	yes	no	yes	yes (only menus/directories)	yes (only menus/directories)

Table 1: Comparison of the features of six different gopher clients.



## The ACN is on the Move Again

### CAMPUS NEWS



Computers, like people, occasionally relocate. In the case of networked systems, this often means that the computer, again like a person, will have a new address. Such is the case for the MVS/CAS system on the Administrative Computing Network (ACN). In mid-December, the TCP/IP interface for the ACN's MVS/CAS system was moved to a router on the FDDI ring at the University of Illinois at Chicago. By connecting this computer to the FDDI ring, AISS hopes to improve the performance of the MVS/CAS system on the network and to increase the number of possible TCP/IP connections it can handle. However, the move also means that the IP address for accessing this system via ftp and telnet will already have changed by the time this article hits the streets. For this reason, AISS encourages all clients to begin using the fully-qualified domain name **UICMVSA.AISS.UIC.EDU** when accessing the MVS/CAS system rather than using its numerical IP address. And, in general, users are urged to always use the hostname as opposed to the IP address, because the IP address might change again sometime in the future, whereas the hostname for this machine will never change.

If you are accustomed to accessing the ACN by typing its IP address when you invoke your TCP/IP applications (e.g., telnet, ftp, etc.), simply substitute **UICMVSA.AISS.UIC.EDU** for the numerical IP address when issuing commands to open a session on the MVS/CAS system. For example, if you ordinarily open a session by typing the command **tn3270 131.193.163.4**, use the command **tn3270 uicmvsa.aiiss.uic.edu** instead.

In addition, all references to the ACN's old numerical IP address (131.193.163.4) in batch files, login scripts, telnet configuration files, and user documentation should be replaced with the host name **UICMVSA.AISS.UIC.EDU**. This step is especially important because, during the transition of moving the ACN's telnet server from the VM/PROFS machine to the MVS/CAS system last year, there was initially a problem accessing the latter system with the host name. Thus, many net-

work administrators and end-users may have quite intentionally configured their software to use the IP address in order to bypass this problem. However, the domain name problems of the MVS/CAS computer have long since been resolved, and now the use of the hostname is desirable, if not imperative.

The new IP address for the MVS/CAS system will be announced in January, but AISS will continue to discourage clients from using it, except in cases where there is a temporary problem with the domain name service. Finally, these changes will in no way affect ftp and outgoing telnet sessions on the VM/PROFS system. Nor will it affect the Internet or BITNET electronic mail addresses of PROFS users. ■

-Lynn Ward

## CCSO Beefs Up Network Support Staff and Services

### CAMPUS NEWS



UIUCnet is no longer just a tool for engineers and computer scientists. Thousands of students, faculty, and staff use the network every day for routine activities. CCSO is increasingly aware of the need to provide high-quality support services to end-users and network administrators. Two new staff members, David Ruby and Steven Hinkle, were hired this fall to augment existing support staff.

David Ruby, a U of I graduate with a major in Spanish and minor in computer science, comes to CCSO from the UIUC College of Liberal Arts and Sciences, where for two years he served as the network administrator for one of the largest and most complex in-building networks on campus—the 400+ node network in Lincoln Hall. While managing the Lincoln Hall site, Dave worked extensively with PCs and Macs on Token Ring, Ethernet, and LocalTalk networks. David is now part of CCSO's User Services group and holds the title Research Programmer/Network Consultant. Currently, he is assisting Randy Cotton, network consultant for the Network Design Office, with a backlog of troubleshooting requests. Dave's responsibilities at CCSO will evolve as needs dictate. It is anticipated that he will be a

primary source for the support and training of network administrators. Dave will also provide information and instruction to CCSO's full and part-time consultants in order to keep them up to date on network applications and technologies.

Steven Hinkle, CCSO's other new hire, will concentrate on the end-user side of networking. Steven, now a Research Programmer/Systems Consultant for User Services, received his B.S. in computer science from the University of Buffalo in August of '92. At Buffalo, while working toward his degree, he also served as a consultant for a variety of computer systems, including PCs and Macs, IBM CMS mainframes, DEC VAX clusters running VMS, and Sun clusters running SunOS. As a newcomer to the U of I and CCSO, Steven is still in the process of acquainting himself with our network, systems, and services. His role in User Services will include offering short courses for end-users on how to use the network, writing end-user documentation, consulting on network-related problems, and supporting popular network applications for PCs and Macs such as *NCSA Telnet*, *Eudora*, *NUPop*, *Trumpet*, etc.

David and Steven make a fine complement to CCSO's existing network support staff, Randy Cotton of the NDO, Declan Fleming and Leslie Rankin (manager and assistant manager of the CCSO sites), and the many CCSO staff members involved with LAN maintenance. Recently, Declan has been working with faculty to make instructional software available at CCSO's networked sites. He also started a Novell Users group, which provides a forum for Novell LAN administrators to share information and discuss specific topics of interest (for more information on the Novell Users group, send e-mail to Declan at [d-fleming@uiuc.edu](mailto:d-fleming@uiuc.edu)).

CCSO systems consultants are also spreading the gospel of UIUCnet. Through a relatively new "outreach" program, consultants are providing on-site seminars to faculty and staff about the services and applications available on UIUCnet, with an eye toward offering special courses on how to use the network for teaching and research. Meanwhile, CCSO managers are taking a fresh look at how to best meet the ongoing network support needs of this campus. Their goal is to develop a flexible, well-rounded program that will maximize the effect of CCSO's support and training efforts. ■

-Lynn Ward

# MacTCP 1.1.1 Available at CCSO Resource Center

## CAMPUS NEWS



This fall, Apple Computer rolled out several new models of desktop and portable computer and, with them, an incremental upgrade of the Macintosh operating system. The new OS, called System 7.1, reportedly offers better font handling, improved stability and performance, improved support for new CPUs (so that system upgrades are not required when new products are introduced), and the incorporation of Apple's multimedia technology known as *QuickTime*.

One thing that System 7.1 does not offer, however, is compatibility with MacTCP versions 1.1 and earlier. MacTCP, Apple's implementation of the TCP/IP protocol suite, provides a standard interface for developing TCP/IP-based software (e.g., *NCSA Telnet*, *Fetch*, *Eudora*, etc.) for the Mac. Virtually every Macintosh connected to the campus backbone has a copy of MacTCP installed in its **System** folder as a control panel device.

To address the compatibility problem between System 7.1 and MacTCP, Apple has released MacTCP 1.1.1. MacTCP 1.1.1 is available free to University staff and students through a product site license. All UIUCnet users who have ordered or received a new Macintosh with System 7.1 pre-installed, as well as users who have upgraded an older Mac to System 7.1, should obtain and install this latest version of MacTCP. The software is available on all of the Macintoshes in the CCSO Resource Center, 1420 DCL. It is located in the AppleShare volume called **RC\_MACs** in the folders **/Public/Communications/Eudora 1.3b109/Mac TCP Software** and **/Public/Communications/Network/Mac TCP/ver 1.1.1**. Remember to bring your own floppy disks (double-sided, double density will do) to copy the files.

According to a press release, in addition to providing compatibility with System 7.1, the new MacTCP offers "better support for extension products, such as AppleTalk Remote Access (ARA), SLIP (Serial Line IP) and PPP (Point-to-Point Protocol) drivers, as well as support for larger 'Hosts' files."

### Installing MacTCP 1.1.1 for the First Time

If you don't already have a copy of MacTCP on your system (as will be the case for individuals who have purchased new Macs), ask your building network administrator to assist you with the installation of the software. He or she can help you determine whether your IP address will be static or server-allocated and whether the address needs to be registered with the UIUCnet *hostmaster* (the person who keeps track of IP addresses and domain names for all systems attached to UIUCnet).

### Upgrading to MacTCP 1.1.1

Mac TCP 1.1.1 is downwardly compatible with earlier versions of the Macintosh operating system and can be installed on Macs running System 6.x, 7.0, and 7.0.1. Before installing the new version of MacTCP, open your existing copy of MacTCP through the Control Panel and record the current settings of the software. First, if there is a LocalTalk icon and an Ethernet icon in the MacTCP window, note which one is selected. If the LocalTalk icon is selected, record the name of the AppleTalk zone that appears beneath it (if a name appears). Then, write down the

number given as the IP Address. Click on the **More...** button and you should see a window that looks like Figure 1 on this page. Write down the information entered in the boxes labeled **Obtain Address**, **Routing Information**, **IP Address** (class and subnet address), and **Domain Nameserver Information**. All of this information must be entered in the new version of MacTCP when you install it.

Once you have recorded this configuration information, you can throw the files named **MacTCP**, **AdminTCP** and **MacTCP Prep** in the Trash (you may, however, want to first copy these files to a floppy disk in case your installation runs amuck). Then, copy the new **MacTCP** and **AdminTCP** files into your **System** folder. For Macs running System 7 or higher, place the files in the **Control Panels** folder within your **System** folder. Once the new files are in place, open MacTCP through the Control Panel and configure the software to match the settings you recorded from your previous version. If you have any problems with the installation or you install it and discover that your network applications do not work, contact your building network administrator for assistance. ■

-Lynn Ward

The screenshot shows the MacTCP configuration window with the following sections:

- Obtain Address:**
  - ☐ Manually
  - ☒ Server
  - ☐ Dynamically
- Routing Information:**
  - Gateway Address:** 128.174.33.1
- IP Address:**
  - Class:** B **Address:** 128.0.0.0
  - Subnet Mask:** 255.255.255.0
  - A visual representation of the IP address and subnet mask using a grid of boxes.
  - Net | Subnet | Node**
    - Bits:** 16 | 8 | 8
    - Net:** 32768 ☐ Lock
    - Subnet:** 0 ☐ Lock
    - Node:** 0 ☐ Lock
- Domain Name Server Information:**

Domain	IP Address	Default
uiuc.edu	128.174.5.58	<input checked="" type="radio"/>
uiuc.edu	128.174.5.50	<input type="radio"/>
uiuc.edu	128.174.5.59	<input type="radio"/>

At the bottom are **OK** and **Cancel** buttons.

Figure 1: The MacTCP configuration window.

# Customizing Telnet Sessions on a PC

## NET TIPS



Although telnet, the TCP/IP remote login application, is probably one of the first pieces of software to which new UIUCnet/Internet users are introduced, very few people have the time or inclination to learn about and exercise all of the options available in the software. In fact, it's probably safe to say that most people on campus don't even have the complete documentation for the version of telnet they use. This is unfortunate. While you may not have time to read the manual from cover to cover, browsing the table of contents or index could expose you to some features that you might find very useful. This month's Net Tip focuses on customizing the telnet environment on a DOS-based PC so that you can easily connect to a specific host without having to remember its full domain name or IP address and preconfigure various session parameters with such hosts. (In the next issue of *UIUCnet*, we will look at the equivalent procedures for the Macintosh.)

### About the Config.tel File

Normally, unless someone has carefully preconfigured your telnet software for you, when you open a remote login session with NCSA or Clarkson Telnet, you must type the name of the executable file that starts the telnet software followed by the full domain name or IP address of the remote host (e.g., **telbin ux1.cso.uiuc.edu**). Additionally, the screen colors, keyboard mappings, scrollbar settings, echo mode, and other parameters will be the same for every session. You can simplify the login process and customize the parameters that govern the sessions with specific hosts by modifying the telnet configuration file called **config.tel**.

The **config.tel** file is a plain text file that contains the default settings used by the Telnet software. The format of the file is quite straightforward. Lines that contain a parameter name followed by an equal sign and a value (in the form **keyword=value**) are read and used by the Telnet software (and related utilities) when opening a session and communicating with a remote host. The meaning of each param-

Table 1: Machine-Specific Entries in a Sample Config.tel File

name=library	#descriptive name for library system
host="garcon.cso.uiuc.edu:625"	#domain name and port number of library system
hostip=128.174.5.58	#IP address of library system
nfcolor=WHITE	#screen color settings for all sessions with library
nbcolor=blue	
rfcolor=red	
rbcolor=cyan	
ufcolor=black	
ubcolor=white	
scrollback=200	
name=ux1	#descriptive/short name for ux1
hostip=128.174.5.59	#IP address for ux1
erase=backspace	#backspace key functions as standard backspace
copyfrom=library	#borrow other parameters from entry named library
name=vmd	#descriptive/short name for VMD
host=vmd.cso.uiuc.edu	#domain name for VMD
keymap=vmd.tbl	#use custom keyboard map file called vmd.tbl for this session.

eter is described in detail in the user's manual and briefly in a comment on the same line in the **config.tel** file. Lines or parts of lines that begin with a pound sign (#) are comments.

The beginning of the **config.tel** file contains important information about the network configuration of your machine and default settings for the Telnet software. Many of these parameters were probably preset by your network administrator (or whoever installed the software), and it's generally best not to fuss with them if your software is working properly. Somewhere toward the middle of the file, however, you should see groups of lines with each group headed by a line beginning with the keyword "name" (e.g., **name=ux1**). Each of these groups describes the parameters to be used when communicating with a specific machine on the network and the keyword **name** is used as a delimiter to separate one machine-specific entry from the next (as well as to provide a short, descriptive name for each system, which can be used in lieu of its IP address or domain name). Generally, the first **name** entry begins with the text **name=default**. This entry contains the default values used for any session

that doesn't have a specific **name** entry in your **config.tel** file and also determines the settings of parameters that are not otherwise specified in a named entry. If you are not satisfied with the default values for screen colors, the scrollbar mode, the default keyboard map, etc., you can change them. Any changes you make will affect all sessions that do not have a specific entry in your **config.tel** file.

Following the default settings for telnet, you should see several other machine-specific entries in your **config.tel** file. Remember, each machine-specific entry begins with the line **name=value** where **value** is replaced with a descriptive name for the system. The parameters following the keyword **name** may be on separate lines, or they may be on the same line, each separated from the next by a semi-colon, colon, or space. In addition to the systems already specified in your **config.tel** file, you can add entries for the machines you access on a regular basis. The example in Table 1 shows three machine-specific entries in a **config.tel** file (with comments)—one for accessing the on-line library system (beginning with **name=library**), one for accessing the CCSO Unix machine ux1

(continued on page 14)



## Customizing Telnet...

(continued from page 13)

(beginning with **name=ux1**), and one for accessing the CCSO IBM mainframe VMD (beginning with **name=vmd**).

Once you have added machine-specific entries to your **config.tel** file, you can access these systems by typing the name of your telnet executable file followed by the value entered for the **name** of the entry. For example, to open a session with the library, you could type: **telbin library**. The Telnet software will look in your **config.tel** file for an entry with the name **library** and use the parameters specified after **name=library** until it encounters the next **name=value** entry in the file—in the case of our example, **name=ux1**. Note that the entry for VMD only contains values for **name**, **host**, and **keymap**. When parameters are not specified in an entry, the values defined in the entry **name=default** will prevail. Thus a machine-specific entry can be very brief, containing only values for the keywords **name** and **host** or **hostip**; or the entry can be quite extensive, containing custom values for every possible parameter.

You can add as many machine specific entries to your **config.tel** file as you like. Most users find it convenient to add the names of systems that they access regularly via telnet or ftp. A list and description of some of the parameters that you can include in each entry are given in Table 2. The list is taken from the **config.tel** parameters for Clarkson Telnet (CUTCP) version 2.2D. If you use NCSA Telnet 2.3, you will notice some slight differences in the way certain parameters are handled. Consult the user's manual for the complete list of machine-specific parameters for each application.

### Editing the Config.tel File

The **config.tel** file can be edited with a plain text editor or word processor. If you use a word processor such as *WordPerfect* or *Microsoft Word*, you must export the edited file to ASCII format when saving your changes, in order to remove any application-specific codes that may have been added to the file by your word processor. Here are some additional pointers to keep in mind when editing the file:

- Before editing any line in your **config.tel** file, be sure to make a copy of the original file with a name like

Table 2: A Partial List of Machine Specific Parameters for the Config.tel File

Parameter	Description
<b>name=value</b>	User-assigned name for system. Replace <i>value</i> with a short name you can remember easily.
<b>host=fully.qualified.domain.name</b>	The domain name for the system you want to reach. Replace <i>fully.qualified.domain.name</i> with the actual domain name of the host you want to contact. If you want to designate a port number as well, separate it from the domain name with a colon and surround the entire value with double quotation marks (e.g. <b>host="garcon.cso.uiuc.edu:625"</b> ).
<b>hostip=###.###.###.###</b>	The IP address of the host you want to contact. If the IP address is not included, telnet will contact one of the <b>nameservers</b> listed in your <b>config.tel</b> file to resolve the domain name given with the <b>host</b> parameter. Including the IP address will generally speed up the time it takes to connect to the specified host.
<b>erase=(backspace or delete)</b>	Determines the function of the backspace key for the session. Some hosts prefer the backspace key to be delete and others prefer the backspace key to be backspace. Set this parameter to <b>erase=backspace</b> or <b>erase=delete</b> .
<b>scrollback=numeric value</b>	Determines the number of lines that can be viewed in scrollback mode. Scrollback uses 86 bytes per line saved. Set your scrollback value to a small number if you are concerned about running out of memory. The typical range is 100 to 200 (e.g., <b>scrollback=200</b> ).
<b>nfc=</b> <i>color</i> <b>nb=</b> <i>color</i> <b>rf=</b> <i>color</i> <b>rb=</b> <i>color</i> <b>uf=</b> <i>color</i> <b>ub=</b> <i>color</i>	The following parameters set the screen colors for the session. Possible colors are black, green, blue, magenta, cyan, red, yellow, and white. When typed in upper case, foreground colors will appear in high-intensity and background colors will blink.  normal foreground color normal background color reverse foreground color reverse background color underline foreground color underline background color
<b>copyfrom=name</b>	Causes the session to use the same parameters specified in another named session in the <b>config.tel</b> file. For example, <b>copyfrom=ux1</b> would cause the current entry to use the same parameters as those specified for the session named <b>ux1</b> unless alternative parameters were explicitly designated.
<b>keymap=filename.tbl</b>	Uses the custom keyboard mapping as specified by the value entered for <i>filename.tbl</i> . Clarkson Telnet provides the user with a default keymap for vt100 sessions and a 3270 keymap for tn3270 sessions. The user can create additional custom keyboard mappings and associate them with a particular session by using this parameter. Otherwise <b>default.tbl</b> will be used for sessions using vt100 emulation and <b>tn3270.tbl</b> for sessions using 3270 emulation.

**config.old** or some other unique name (do not use the name **config.bak** because your text editor may use it as a default backup name, and you could potentially overwrite your original during the editing process). Taking this precaution will enable you to resurrect your original telnet configuration in the event that your modifications cause Telnet to malfunction.

- Do not modify or delete any line in the **config.tel** file if you are uncertain about its function.
- In the machine-specific section of the file you are likely to encounter entries with parameters such as **nameserver=#** and **gateway=#**. These entries have probably been preconfigured by the

(continued on page 15)

# Navigating and Using the Internet: A Hands-on Course with a Heart

## CAMPUS NEWS



A new course will be offered through the Graduate School of Library and Information Science this spring: **LIS450CC, Advanced Problems in Librarianship—Telecommunications**. The course combines the learning of real-world skills for navigating the Internet with a survey of the major uses of and issues related to the Internet. Professor Greg Newby will teach the course. Newby, who presented a similar course at Syracuse University, also hopes to offer LIS450CC to students at remote networked sites through UIUC's extramural program in the fall of '93.

Professor Newby's course will concentrate on computer networking as a medium for human communication. Familiar forms of human communication—the telephone, postal service, and face-to-face interaction—will be compared and contrasted with networked communication. Computer networking shares qualities with traditional communications media, but also has important differences. The norms and standards of networked communication are not yet well established, and the channels of communication—for example, ftp, mailing lists, and various

information services—are not always easy to understand and/or access.

A thread that runs throughout the course is the rather gray area of what constitutes acceptable and unacceptable behavior on the Internet. For spoken communication and the printed word, there are both well-defined social rules for interaction and a body of law to deal with transgressions. In the datasphere, however, few laws exist and the norms of conduct are still being formed. A class session will be devoted to topics such as the Morris Internet Worm and the West German hacker who was caught by Cliff Stoll. These and other examples will lead to discussions about how the end-user can better recognize and be prepared to deal with situations requiring ethical judgment within a networked environment.

During the first seven sessions, students will spend about half of each class in a computer lab acquiring hands-on experience. The Unix operating system, elec-

tronic mail, and other basic network tools such as telnet and ftp will be covered in detail. Students will also have the opportunity to investigate more specialized applications, such as Gopher, the World-Wide Web (WWW—a hypertext application used for locating and accessing information on the Internet), WAIS, Archie, and Hytelnet (a hypertext application for facilitating telnet sessions with on-line library catalogs and information servers).

Throughout, the focus is on using network-based resources to extend each student's personal and professional information resources. Students will be encouraged to identify ftp sites, databases, or mailing lists and newsgroups containing information pertinent to them. They will also be encouraged to interact with their peers or experts in various fields by using network tools, and to consider the electronic dissemination of their thoughts, findings, and papers.

The balance of the course is a survey of various perspectives on computer networking. First, a foundation in human communication theory and practice will be laid, and, from there, various aspects of computer networking will be addressed. The history of networking and many types of computer networks and protocols for data transfer will be treated in detail. Corporate communication, information storage and retrieval services, and public access to networked services will also be covered. Finally, the course will address questions about the future of computer networking—e.g., how NREN (the National Research and Education Network) and an increased network presence in libraries and K-12 schools might change the users and uses of Internet.

LIS450CC will meet on Mondays from 12:00–3:00 P.M. during the spring semester. Non-LIS grad students and auditors are invited to attend, space permitting. Many of the readings for the course as well as the syllabus may be obtained via anonymous ftp from the host **gpx.lis.uiuc.edu**. These materials are located in the directory **pub/netinfo**. Specific questions about the course can be directed to **gbnewby@alexia.lis.uiuc.edu**.

—Greg Newby



## Customizing Telnet...

(continued from page 14)

person who installed your telnet software and should not be changed unless you are certain that they are incorrect. The **nameserver** entries instruct telnet to ask specific machines on the campus net to translate domain names into IP addresses. The **gateway** entry specifies the address of the machine that connects your building network to the campus backbone.

- If a comment wraps around to a second or third line, be sure that each line begins with a pound sign (#).

- Even if you do not have a copy of the entire Telnet user's guide, have the section pertaining to the **config.tel** file on hand for reference as you make your changes.

- Finally, editing a **config.tel** file for the first time can be a little intimidating. If you are uncertain about what you are doing, your building network administrator can probably provide some assistance. Additionally, the CCSO micro-computer consultants are well-versed on Clarkson and NCSA Telnet and can be reached at 244-0608. ■

—Lynn Ward



1120 DCL  
Campus MC 256  
1304 W. Springfield Ave.  
Urbana, IL 61801

VINCENT GOLDEN  
SPECIAL COLLECTIONS ADMIN OFC  
314 LIBRARY  
CAMPUS  
MC-522

## Gopher Clients...

(continued from page 10)

The general syntax of the line should be:

```
set g_tel=c:\telnet_directory\ telnet_app -h  
c:\telnet_directory\ config.tel %%a %%p
```

where *telnet\_directory* is replaced by the name of the directory in which your telnet program is stored and *telnet\_app* is the name of your telnet program's main executable file (e.g., *telbin.exe*, *tn.exe*, etc.) When telnet is invoked by PC Gopher II, the parameters *%a* and *%p* are replaced with the IP address and port information provided by the gopher server (if you type the *set* command at the DOS command prompt instead of including it in a batch file, use a single percent sign in front of the *a* and *p* parameters [e.g., *%a %p*]). When using PC Gopher II in conjunction with Clarkson and NCSA telnet, it may be necessary to surround the *%%a %%p* parameters with double quotes in order for telnet to properly recognize a port number. For example, the *g\_tel* variable in my *autoexec.bat* file looks like this:

```
set g_tel=c:\telnet\telbin.exe -h c:\telnet\  
config.tel "%%a %%p"
```

PC Gopher II is a memory hog and does not support or display tn3270 sessions, so there is no reason to use the larger application *tn3270.exe* as the telnet program file.

There is still much work to be done before PC Gopher II can compete with the clients mentioned above. This client only recognizes, displays, and transfers ASCII

text files. Thus, file transfers are extremely limited and archie queries produce unpredictable results. Also, occasionally binary files (such as GIF files) are misconstrued as text files and PC Gopher II tries to display them. Additionally, the memory constraints of PC Gopher II prevent it from displaying the complete text of even moderately sized files. Such files must be saved to disk and read or printed with a text editor or word processor. Hopefully, many of these problems will be ironed out in the next incarnation of this DOS client. In the meantime, if you have a Unix account, you may prefer to use the Unix curses client instead.

### CMS Gopher 2.2.0

CMS Gopher 2.2.0 is quirky, to say the least. On the one hand, it recognizes and displays the names of almost all object types. However, it can only display and transfer plain text files. To save a text file, the file must first be loaded into the XEDIT editor (there is no *save* command in gopher itself). Although the client supports ph and archie searches, with a few rare exceptions index searches fail and return an error message. The most ironic shortcoming of this client developed for IBM mainframes is that, while it can launch telnet sessions and communicate clumsily with Unix hosts, it does not support tn3270 sessions with other IBM mainframes. An attempt to launch such a session will result in the error message "Can't process this file type." A new version of CMS Gopher is available, but as of this writing, has not been installed on VMD. Perhaps some of the oddities just noted are corrected in release 2.3.

## For More Information

For the most up-to-date information about gopher and the gopher software distribution, browse the documents and the **Gopher Software Distribution** menu under the **Information about Gopher** menu on UIUC's main gopher server. ■

-Lynn Ward

## About UIUCnet

UIUCnet provides timely information about the campus network. It is published by the Computing Services Office and is edited by Lynn Ward. Comments and suggestions for topics are welcome. Permission to reprint all or part of UIUCnet for non-profit purposes is granted, provided full acknowledgement of the source is given. Feel free to reach us via electronic mail ([uiucnet@uiuc.edu](mailto:uiucnet@uiuc.edu)), campus or U.S. mail (UIUCnet, Computing Services Office, 1120 Digital Computer Laboratory, MC 256, 1304 W. Springfield Ave., Urbana, IL 61801), or by phone at (217) 244-0681.

For a free subscription, just send us your name and (preferably campus) address; be sure to include your department and mail code if you send a campus address. UIUCnet is also available in electronic form. Recent issues are posted to the Netnews newsgroup [uiuc.pubs.uiucnet](mailto:uiuc.pubs.uiucnet). Back issues are available for download from the anonymous ftp host <ftp.cso.uiuc.edu> in the directory [doc/net/uiucnet](ftp://doc/net/uiucnet) and from the UIUC Gopher server.



004160977366  
U17  
6:2 1993 COPY 2

STX

# UIUCnet

The University of Illinois Campus Network

Vol. 6 No. 2

Feb. - Mar. 1993

## Icon Key

### Audience



Novices



Experienced users



Network Administrators



AISS clients



Everyone

### Platform/Operating System



PC compatibles (DOS™)



Macintosh™



X Window System™



NeXT™



CCSO'S IBM™  
mainframe, VMD



UNIX™ mainframes  
and workstations



Any platform

## Free-Nets: Letting the Network Out and the Community In

### MISCELLANEA



There is a certain irony about being a member of the Internet community. With a few keystrokes, you can deliver a message to a friend or colleague halfway across the globe; but sending electronic mail to a neighbor equipped with a computer and modem may be impossible, unless he or she happens to be a student or employee of the University. In the fall of '92, anyone on the Internet could download or read online the speeches and position papers of U.S. presidential candidates Bill Clinton and George Bush, but to obtain comparable information on seekers of local and state offices we were forced to turn to more traditional media. And, even when information of local interest is available on the Net, it may not be accessible to the audience who could most benefit from it. For example, students and staff at the University of Nottingham, UK (and hundreds of other sites around the world), can read the enthusiastic endorsement of an Urbana veterinarian or scan the spring schedule of the Krannert Center for the Performing Arts by linking to the UIUC Gopher server. Yet, well over half the population of Champaign county is unable to make use of this valuable resource. Simply put, the numerous resources on the Internet are currently only available to community members fortunate enough to be as-

sociated with an institution that has Internet connectivity (or wealthy enough to pay for their own connection). Fortunately, this situation will change in the very near future. Members of the UIUC Graduate School of Library and Information Science are spearheading a movement to establish a *Free-Net* in Champaign-Urbana.

### What is a Free-Net?

A Free-Net is a community owned and operated online information system, available free of charge to anyone with access to a computer and modem. The Free-Net concept originated with Dr. Tom Grudner, formerly a staff member of the Department of Family Medicine at Case Western Reserve University. In 1984 Grudner set up a small electronic bulletin board to allow health professionals to answer questions and deliver general health information to the public. The project was so successful that—with the financial backing of AT&T, the Ohio Bell Telephone Company, and University Hospitals of Cleveland—Grudner transformed his single user



(continued on page 2)

## Free-Nets...

(continued from page 1)

BBS into a comprehensive multi-user community information service known today as the Cleveland Free-Net. The first version of the Cleveland Free-Net opened its ten dial-up ports to the public in 1986, and, like its progenitor, provided a venue for lay people to get information and advice from professionals in the Cleveland area. Seven years and several incarnations later, the system now boasts over 350 separate information and communications features (including many Internet services) and handles up to 12,000 logins per day.

In 1989, Grudner founded the National Public Telecomputing Network (NPTN), a non-profit agency dedicated to helping other communities develop Free-Nets and delivering high-quality network broadcasting services (called *cybercasts*) to NPTN affiliates. There are now at least 11 active Free-Nets scattered throughout the U.S. Canada, and New Zealand with many more scheduled to come online this year, including *Prairienet*, the Champaign-Urbana Free-Net.

### What Can I Find on a Free-Net?

Every Free-Net has its own personality, reflecting the interests and dedication of the community that supports it, but there are common characteristics among the systems. First, most Free-Nets use one of two commercially available software packages, *Freeport* or *Community Information Exchange (CIX)*. Both products run on Unix systems and offer a user-friendly front end to many different kinds of services. The Free-Net interface uses the metaphor of an electronic city. Each item on the main menu takes the user to a different *location* in the city, and each location offers different services (see Figure 1). For example, the Post Office is a place to send, receive, and manage electronic mail. The Library may offer access to local and distant electronic library catalogs as well as electronic books and newspapers. Users can go to the Public Square to engage in real-time interactive conversations with other system users or vote on community issues. Free-Net menus can be totally customized, so ultimately it is up to the system administrator to design an electronic city that guides the user intuitively to the specific services available on the system.

Even though services vary from system to system, most Free-Nets offer a set of core services. These include:

- **Internet Connectivity.** Most Free-Nets are affiliated in some way with a college or university. If the Free-Net system is connected to the campus network and the campus network is connected to the Internet, standard TCP/IP applications (telnet, ftp, etc.) may be available. It is also possible to install other popular network applications such as Gopher, HYTELNET (an application for accessing remote databases and library catalogs), and WAIS. These services are, of course, especially beneficial to community members who wouldn't otherwise have access to them

- **Electronic mail.** Any registered user can exchange e-mail with any other user on the system. If the Free-Net has Internet connectivity, users can send mail to and receive mail from anyone on the Internet and possibly other networks such as BIT-NET, CompuServe, GENIE, etc.

- **Usenet Network News.** The FreePort software comes bundled with Network News server (NNTP) software, the Netnews transport software C-NEWS, and a simple, menu-driven news reader. Thus, all users can read and contribute to a selection of world-wide Usenet newsgroups.

- **Special Interest Groups.** Special Interest Groups, or SIGs, are unmoderated discussion groups on topics of local interest. Any registered user can ask the Free-Net system administrator to create a SIG. SIGs look and behave like Usenet newsgroups, but they are not exported to other Usenet sites, so most or all participants are geographically proximate to one another. It can be fun to interact with a fellow angler from Cambridge via the *alt.fishing* Usenet newsgroup; but a Free-Net Fishing SIG offers the possibility of establishing a "real-

time" friendship with someone across town who knows the best local fishing holes.

- **Moderated Discussion Groups and Sponsored Information Areas.** Free-Nets are first and foremost community information systems. The success of the system often rests on the willingness of local professionals and organizations to sponsor information areas. For example, local doctors, lawyers, auto-mechanics, accountants, agronomists, etc. could upload electronic brochures and fact sheets and/or answer specific questions posed by users (Q&A services normally take the form of moderated newsgroups); the minutes of city council or school board meetings could be posted by a committee member; government officials could interact electronically with their constituency. The possibilities for distributing information both rapidly and creatively are almost limitless.

- **Cybercasting Services.** Through a special arrangement with American Cybercasting Corporation, NPTN provides all affiliates with cybercasting services, high-quality information features on a variety of subjects. The multi-faceted *Teledemocracy Project*, for example, provides users with 1) the full-text of Supreme Court Decisions, 2) a weekly summary of three U.S. Senate and three House bills with information on how representatives voted on each measure, 3) a congressional contact file with the names, addresses, phone numbers, and committee assignments of federally-elected officials, and 4) occasional short-term features such as the Campaign '92 information service. Besides free cybercasting services, fee-based subscriptions to electronic news services, journals, and magazines can be made available to ALL registered Free-Net users (in which case, the Free-Net picks up the tab), or to a select group of subscribers who are billed periodically.

- **K-12 Educational and Recreational Programming.** Free-Nets provide community youths with the opportunity to explore the vast potential of telecomputing. NPTN's Academy One cybercasts offer programs for students, parents, educators and school administrators. Interactive programs such as simulated spaces shuttles, journeys to "virtual worlds," and the "TeleOlympics" use Free-Net's electronic mail and conferencing features to allow participants to interact with one another and coordinate these world-wide

(continued on page 3)

#### <<< CLEVELAND FREE-NET DIRECTORY >>>

- 1 The Administration Building
- 2 The Post Office
- 3 Public Square
- 4 The Courthouse & Government Center
- 5 The Arts Building
- 6 Science and Technology Center
- 7 The Medical Arts Building
- 8 The Schoolhouse (Academy One)
- 9 The Community Center & Recreation Area
- 10 The Business and Industrial Park
- 11 The Library
- 12 University Circle
- 13 The Teleport
- 14 The Communications Center
- 15 NPTN/USA TODAY HEADLINE NEWS

h=Help, x=Exit Free-Net, "go help"=extended help

Figure 1: The Cleveland Free-Net main menu; an electronic city.

## Free-Nets...

(continued from page 2)

events. The NPTN Student News Network distributes electronic versions of school newspapers. Academy One's Daily Report Card is an eight page briefing for educators on "America's progress toward better schools." A special NPTN news-group called the Educator Contact File facilitates communication among teachers who are interested in coordinating telecomputing projects. A special area of the net might also be reserved for teen or youth SIGs and other recreational services.

These are just a few of the many items you'll encounter on a Free-Net. To really get a feel for what a Free-Net is like, try logging in to a remote Free-Net as a guest with telnet. See the table on this page for a list of Free-Nets (including the C-U prototype), their domain names, and guest login IDs.

### Who Pays for Free-Net Services?

Although there is no charge to get a registered account on a Free-Net, monies are needed to purchase core equipment (a multi-user server, phone lines, modems, etc.) and the staff to run it. Also, in order for the Free-Net to become a viable community resource, there is the additional cost of providing public access terminals in schools, libraries, and other places where people gather. Like public television and radio, Free-Nets rely heavily on corporate endowments and private donations. Regular users, for example, may be asked to contribute annual, but strictly voluntary, dues.

### About *Prairienet*: the C-U Free-Net

The Champaign-Urbana Free-Net project, dubbed *Prairienet*, has been under way for roughly a year. Actively involved are members of the Graduate School of Library and Information Science (Greg Newby, Ann Bishop, Leigh Estabrook, and Marie Erdman) as well as Mike Gardner of CCSO, David Micko (former GSLIS student), and Champaign resident Greg Smith.

Greg Newby has drawn up a two-year implementation plan, and although things are slightly behind schedule, real progress has been made. Free-Net presentations and demos have been given at community and school libraries. Tom Grudner of NPTN arranged for an Ameritech grant to cover the cost of the Free-Port software for *Prairienet*. Newby and Erdman installed the software on Newby's Unix workstation and have set up a working prototype.

In order to go public with the system, the organizing committee is actively soliciting

funding (and other types of support) to acquire basic hardware (a dedicated, mid-range Unix workstation; 32 phone lines, modems, and a terminal server; and a collection of terminals and modems to be placed at public access sites) and to hire a full-time director and half-time systems administrator. The *Prairienet* director would ultimately be responsible for fund-raising, public-relations, and enlisting the aid of local information providers (professionals and organizations) to sponsor community-based information areas on the system. A number of UIUC departments interested in using *Prairienet* to disseminate information have already pledged sponsorship if matching funds can be raised in the community.

In its start-up phase, *Prairienet* will offer Champaign-Urbana residents services such as electronic mail, a large Usenet news feed, Special Interest Groups, community information services, and links to various Internet facilities. The system will support 30-40 simultaneous sessions and roughly 2000 registered users. Subsequent phases of the project involve the expansion of hardware, information services, user capacity, and staff.

*Prairienet* will be initially housed on campus and will make use of existing campus computing services rather than duplicating them. From the start, though, *Prairienet* will have a focus on community information and

will seek to engage community members and organizations in the creation and maintenance of the various information and activity areas to be found there.

### Volunteers Welcome

*Prairienet* offers the Champaign-Urbana area an opportunity to break down the communication barriers between the "have nets" and the "have nots." University staff and students, who already have Internet access, can take advantage of *Prairienet*'s community-oriented services. In addition to community information services, community members will also have free access to Internet applications, making it possible for them to communicate electronically with people at the University or anywhere on the Internet.

If you are interested in volunteering time, money, or expertise (perhaps you'd like to sponsor an information area) to *Prairienet*, send an e-mail message describing your intentions to [prairienet@uiuc.edu](mailto:prairienet@uiuc.edu) or paper mail to Greg Newby, 410 DKH, MC 707, 1407 W. Gregory Dr., Urbana, IL 61801. Also, old terminals, obsolete PCs, and slow modems can make fine public access terminal equipment. If you own computer equipment (privately, not University inventory) that no longer serves your needs, this too can be donated to the *Prairienet* cause. ■

- Lynn Ward

### A Partial List of Free-Nets

Name / Location	Domain Name	Login / Password
Prairienet (prototype) Champaign-Urbana, IL	gpx.lis.uiuc.edu	visitor / guest
Heartland Free-Net Peoria, IL	heartland.bradley.edu	bbguest
Youngstown Free-Net Youngstown, OH	yfn.ysu.edu	visitor
Lorain County Free-Net Elyria, OH	freenet.lorain.oberlin.edu	guest
Cleveland Free-Net Cleveland, OH	hela.ins.cwru.edu	visitor
Denver Free-Net Denver, CO	freenet.hsc.colorado.edu	guest
Tallahassee Free-Net Tallahassee, FL	freenet.fsu.edu	visitor
Victoria Free-Net Victoria, B.C., Canada	freenet.victoria.bc.ca	guest
National Capital Free-Net Ottawa, Canada	freenet.carleton.ca	guest



# Eudora 1.3 Released

## Supports Terminal Server Login

### CLIENTS FOR CLIENTS



In early February, Steve Dorner put the finishing touches on the long-awaited official release of *Eudora 1.3* for the Macintosh. Besides offering many new features that improve the overall functionality of the software, this release should be of special interest to modem users because it supports the new login procedure for the CCSO terminal servers.

#### New Features in Eudora 1.3

Many of the improvements in Eudora 1.3 came about as a direct response to end-user feedback. A file called **1.3Changes**, available with the software distribution, briefly describes all the features that have been implemented since the last official release, Eudora 1.2.2. Several of the more enticing new options are highlighted below:

- **Subject Editing.** Have you ever received an e-mail message with a blank or ambiguous **Subject:** field in the message header? If you save the message in one of your mailboxes, it may be difficult to remember what it was about without actually opening the message. Eudora 1.3 allows you to edit the subject of any incoming message. To use this feature, turn on the Icon Bar with the **Switches...** option under the Special menu. When the Icon Bar is on, you can enter/edit any text in the **Subject:** field that appears above the double line of an open message. Editing the subject in this manner does not change the contents of the original message header, but it does change the subject line in the mailbox window.
- **Message Priorities.** You can assign a priority to both outgoing and incoming messages. Like subject editing, this function is only active when the Icon Bar is turned on. Clicking on the small box in the left hand corner of the Icon Bar opens a pull-down menu that lets you assign one of five priorities from "highest" to "lowest" to any message. For outgoing messages, the priority will not affect the speed with which your message is delivered, but it will let the recipient know how important you think the message is. Assigning priorities to incoming messages can serve to remind you about the relative urgency of messages in a given mailbox window. Special priorities on incoming messages are displayed in the Icon Bar of open messages and the Status column of mailbox windows.
- **Make Nickname Command.** With the previous version of Eudora, creating nicknames required either entering both the nickname and full e-mail address manually or doing some creative copying and pasting. The **Make Nickname** command under the Special menu can create nicknames automatically from the addresses in the **To:**, **Cc:**, and **Bcc:** fields of selected outgoing messages, or the **From:** field of incoming messages. You can also select multiple nicknames from the Nicknames window to create a group nickname with this option.
- **Finger Client.** The **Ph** option under the Special menu now includes a *finger* client. Finger is a utility that produces detailed information about a user account. It is especially useful when you know someone has an account on a particular machine, but his/her e-mail address is not listed in the CCSO Nameserver. The syntax for a finger search is:  
  
*name@fully.qualified.domain.name*

where *name* is either the first name, last name or login name of the person you want to look up. For example, suppose you have a friend named Margaret who has an account on *uxa*, but you can't remember exactly how to spell her last name and you don't know her e-mail address. You could open a Ph window in Eudora and enter the following information in the query box: **margaret@uxa.cso.uiuc.edu**. Next click on the **Finger** button. The finger server on *uxa* will return the names of every user on *uxa* whose real name contains the string "margaret". The output will look something like this:

```
Login name: margie8      In real life: landers margaret m
Directory: /students/home1/margie8 Shell: /bin/csh
Last login Wed Dec 9 19:30 on ttysR from or-mac09
No Plan.
```

```
Login name: msg1407     In real life: Margaret Sarkissian
Directory: /students/home1/msg1407 Shell: /bin/csh
Last login Fri Feb 12 05:49 on ttyqZ from mossberg
No Plan.
```

```
Login name: mdf31042    In real life: Margaret D. Ford
Directory: /students/home0/mdf31042 Shell: /bin/csh
Never logged in.
No Plan.
etc.
```

If the person you were looking for was Margaret Sarkissian, you could send her mail by placing her login name, "**msg1407**", before "**@uxa.cso.uiuc.edu**", i.e., **msg1407@uxa.cso.uiuc.edu**.

- **Reply All Switch.** Eudora can now be configured so that it will reply to all recipients of a message by default. Previous versions of Eudora addressed replies to the sender only, unless the option key was depressed when making the reply. The **Reply All** switch is found in the **Switches...** option under the Special menu. Another switch called **Include Self** will add your name to the list of recipients when **Reply All** is turned on.

(continued on page 12)

# Ethernet Addresses: Six Little Numbers That Could Save Your Network

## ADMINISTRIVIA



*Administrivia is a new UUCnet column contributed by David Ruby, network consultant for CCSO's User Services group. While all are invited to read the column, it is specifically directed toward the often overworked and underpaid people who keep our packets flowing—departmental and building network administrators.*

**H**ow do you keep track of where everything is on your network?" somebody once asked me. This simple question can give even the best network administrator the shakes. The best way to keep track of where things are on a network is to maintain an organized written log or a database of important network information. But often networks start out small. You don't start with a network of 400 nodes, you build up to it. Who needs to document anything when your network is made up of three workstations and a printer, right? Unfortunately, all that wonderful "free time" you had before your baby network grew into a monster is used up with learning things like how to make the network work, and not in organizing. Most net admins don't have time to think about documenting the network until after it's almost finished. If you're really lucky, the network gives you a few days to look at your creation before it comes crashing down. I'm going to try to pass on a few things you can do during those mythical no-network-problems days that will make those all too familiar network-problem days a little less traumatic.

One of the many important pieces of information you should keep about your building-wide network (or even a small departmental LAN) is the Ethernet address of each node. An Ethernet address is a unique combination of six hexadecimal numbers, which is encoded into the firmware of every Ethernet card. What's so important about this number? Pretend your network is a street, and the nodes are houses. When nodes send packets to the

network, it's just like people sending packages of information to each other along the street. In order for a package to get to the right house, the correct address must be used. Each machine on the network is identified to the rest of the network by its unique Ethernet address, just as the address on a house is what identifies it to whomever is walking down the street. If your network is healthy, you'll rarely have to deal with things at the network address level, but when something goes wrong, those numbers could be your lifeline.

## Sniffing Packets

If you don't have one yet, purchase or download a *packet sniffer*. Packet sniffers are programs that capture the packets flying by on your network. They don't interfere with the packets—they just copy them down so you can actually see the traffic on your network. Much of what you'll see is just numbers and other data that are only meaningful to your Ethernet cards. However, each packet will have a few bits of information that will always be useful. There will be a destination address and a source address. There also will be a few numbers that specify the protocol in use. If you're trying to see what happens when you send information to a network printer, you could watch all the packets that have the printer as the destination address with your sniffer. If you're trying to figure out why telnet is slow, you'd want to watch all the packets that are using the TCP or IP protocol. Of course, to make anything of all this information, you'll need a list of the physical location of every node on your network along with the Ethernet address of each node.

To demonstrate the usefulness of packet sniffers, let's suppose you have a building full of Macs and PCs with Ethernet cards. Users happily telnet out of the building, connecting to various servers on and off campus. All in all, the network is a fully functional roadway to the world and people are happy. Then, something goes horribly wrong. An Ethernet card somewhere on the net has fried a chip and won't shut up. It seems to be stuck in the transmit mode and is flooding the network with

meaningless packets. No other node can get a word in edgewise. Your phone starts ringing off the hook and your pager is beeping mercilessly with messages from unhappy users. Your packet sniffer helps you determine that a card with the address of **00 00 C0 A1 03 1D** is having a nervous breakdown and needs to be turned off. Right about now, you realize that you don't have a clue about the location of the machine with the address of **00 00 C0 A1 03 1D**. Having a list of which cards are where could save the day, which returns us to our theme: "document your network!"

A network crisis doesn't have to be as dramatic as the whole building network shutting down. A commonplace problem is a "stolen IP address." If you have a building full of people telnetting and ftping, you're probably aware of the need to keep IP numbers unique. If two people try to use the same number simultaneously, neither one of them will be able to telnet or ftp. This is a particularly annoying problem because it's usually a case of an error in a configuration file, so nobody knows they're stealing someone else's number. It's also a difficult problem to track down because it only occurs when both people are trying to telnet at the same time. If one user works in the morning and the other in the afternoon, you'll never hear about it until the one weekend they both come in to catch up on their work. By capturing all the packets with the IP number in question with your packet sniffer, you can isolate the Ethernet addresses of the competing machines. If you have a list of where each address is physically located, this problem can be corrected very quickly.

## Documenting Ethernet Addresses

The easiest way to document an Ethernet address is to write down the address before you install the card into a computer. On most Ethernet cards, the address is marked on the card. Often the address is on a sticker on a chip, but it also can be printed on the board itself. Unfortunately, it's not marked on the part of the card you can see once it's inside the machine. To find out the address of a card that has

(continued on page 6)

## Ethernet Addresses...

(continued from page 5)

already been installed, you may have to open the machine, and take the card out—hardly something efficient to do during a building-wide crisis. One simple suggestion, which may save many headaches later, is to record the address in permanent marker on the part of the card that shows once it's installed. If space is limited, try to get the last three numbers written on the outside of the card (the first three numbers aren't as important, a point we'll get into later). Note that labeling the machine itself isn't necessarily helpful. You can move Ethernet cards from machine to machine, so unless you're careful, there is no guarantee that the same card will be in the same computer the next time you need to know its address. Along the same lines, if you do move an Ethernet card from one machine to another, remember to update your network log or database.

What do you do if you have a few hundred machines with cards in them already? Do you have to open them all up? Well, no. There is plenty of software that can help you here. On a PC, you can use the diagnostic software that came with the card. The diagnostic software will typically give you the Ethernet number as well as information like the card's hardware interrupt, base RAM address, base I/O address, and even the slot you put it in (if it's a micro-channel or EISA machine), all of which can be very useful for troubleshooting later. If you use packet drivers on your PCs, you can also find out an Ethernet address by watching the driver as it loads. When the packet driver software is executed, the address of the card is echoed to the screen.

Finding the Ethernet address for a Mac is easy if you are using a recent version of *MacTCP*. Go to the *MacTCP* module in your control panel. Notice that you've got two or three choices of networks (*LocalTalk*, *Ethernet*, and perhaps *EtherTalk*). Click on the *Ethernet* icon while holding down the option key, and the address of the Ethernet card will magically appear. But what if you're not using *MacTCP* or have the older version? You can download a free utility called *Node Informer* that does the same thing as the *MacTCP* trick and allows you to save the information to a file. To retrieve *Node Informer*, ftp the file called *Node\_Informer\_1.1.sea.hqx* from the **pub/**

**Net\_Tools** directory at [ftp.cayman.com](ftp://ftp.cayman.com). Also, some commercial packet sniffer packages include programs that will collect some Mac information (like chooser name, or the name of the hard drive, and the type of Mac in use) along with the network address. In a limited area, you could determine the network address of each machine by using this tool alone, but it's still a good idea to go and verify the addresses yourself.

### More about Ethernet Addresses

Let me conclude by providing a little background on Ethernet addresses. First, the numbers in an address are in hexadecimal (base 16) notation. The value in each column can range from 0 through 15; thus, we need sixteen different symbols. The first ten digits use the traditional decimal symbols 0-9. For the six additional digits, the letters A through F are used (i.e., A=10, B=11, C=12, and so on). The hexadecimal number **0F** equals 15 in decimal. **AF** is 175 ((10x16)+15) in decimal. It's important not to ignore the letters when you write down an Ethernet address because they are as much a part of the address as the numbers. The first three numbers (**00 00 C0** in my earlier example) indicate who made the card. Every manufacturer of Ethernet cards

is given a certain prefix (or set of prefixes depending on how many cards they've made), which makes up the beginning of the address. In my example, **00 00 C0** tells us that Western Digital made the card and, therefore, it's a PC card as opposed to a Mac card (since Western Digital doesn't manufacture Ethernet cards for Macs). A semi-current listing of these prefixes can be found in RFC 1340, a document that provides various types of Internet address information. (RFC 1340 can be retrieved via anonymous ftp from [uxc.cso.uiuc.edu](ftp://uxc.cso.uiuc.edu). This document is located in the **RFC** directory and has the filename **1340**.) You can usually figure out the first three numbers of an Ethernet address by knowing which company has which prefix. This is why recording the last three numbers is more important if you only have room to write down half of the address on the card itself. The only problem is that the list of Ethernet prefixes is always behind the times. In short, there is no substitute for keeping a written or electronic log of critical network information. ■

- David Ruby (Note: If you have questions about packet sniffers or other technical aspects of this article, contact Dave Ruby via e-mail at the address [daver@uiuc.edu](mailto:daver@uiuc.edu).)

## Network Design Update

### NEW UIUCNET BACKBONE CONNECTIONS: (since 9/92)

909 S. 6th - Div. of Mgmt. Info.  
Architecture  
Coble Hall - Graduate College  
Coordinated Science Lab  
DCL - Workstation Lab  
EASB - National Soybean Research Lab  
EERL - University Police  
Flagg Hall  
Geological Survey Lab  
Harker Hall - UI Foundation  
Housing Food Stores  
Inst. of Government and Public Affairs  
Krannert Center  
Mailing Center  
Main Library  
Observatory  
Rehabilitation Center  
Stadium  
University Fire Station  
University Press

### NETWORK INSTALLATION IN PROGRESS:

Allerton House  
BMRL/College of Medicine  
Center for Advanced Studies  
English - LAS Development Office  
ISR Residence Halls  
McKinley - Counseling Center  
Sheffield Vivarium  
University High School

### NETWORK INSTALLATION ANTICIPATED:

805 W. Penn. - Educ. Theory Annex  
Agr. Bioprocess Lab  
Burnsides Research Lab  
Fire Services Institute

### NETWORK DESIGN IN PROGRESS:

911 S. 6th - AISS  
Assembly Hall  
Campus Bookstore  
Ceramics Bldg.  
Chemical and Life Sciences Bldg.  
Davenport Hall - Geography  
Grainger Engineering Library  
Illini Hall - Math  
Illini Union  
Law Bldg.  
Mumford Hall  
Noble Hall  
Plant Sciences Lab  
Roger Adams Lab  
Temple Buell Architecture Bldg.



## Customizing Telnet Sessions on a Macintosh

### NET TIPS



Last issue's **Net Tips** column focused on how to customize NCSA and Clarkson Telnet for the PC by modifying the **config.tel** file. Although NCSA Telnet for the Macintosh has a **config.tel** file that can be edited in a similar manner, there is a much simpler way to configure telnet sessions on the Mac. To learn how, read on.

### Saving Telnet Configuration Sets

There are two ways to open a telnet connection using *NCSA Telnet* for the Macintosh. The first time you connect to a host, you must use the **Open Connection...** option under the File menu. You'll be prompted to enter the Session Name (domain name or IP number) of the remote machine and optionally to provide a name for the session window. You can also click on the **Configure** button and change many default parameters. Once you click on the **OK** button, assuming that you typed in the correct hostname or IP address, the session window will appear and you can log in to the specified host. This is the most common method for logging in to a host and it works perfectly well. However, if you log in to the same host or group of hosts every day, you can avoid typing the session name and setting the desired parameters each time you log in by creating a telnet *configuration set*. A telnet configuration set is a separate file containing all of the configuration information (hostname, window name, scrollbar setting, backspace/delete setting, window colors, fonts, etc.) about a particular telnet session. Once a set has been saved, you can connect to the host using the identical settings by simply double clicking on a configuration set icon.

The procedure for saving a telnet configuration set is given in the **Advanced Features** section of the Telnet user's manual, but it is actually quite easy and can be as useful for novices as old pros. Here are the steps involved:

1. Start the Telnet software by double clicking on the telnet application icon.
2. Select the **Open Connection...** option under the File menu. A dialog box will

prompt you to enter a Session Name and a Window Name. For the Session Name, enter the fully qualified domain name of the host to which you want to connect. For example, if you wanted to connect to the CCSO computer ux1, you would enter: **ux1.cso.uiuc.edu** (or you could enter the IP number of the host instead, but domain names tend to be easier to remember). For the Window Name, enter the name that you would like to appear on the title bar of your session window. If you do not enter a Window Name, telnet will provide a default name.

Now click on the **Configure** button. A menu of configuration options will appear. You can either accept the current defaults or change them (an explanation of each option is given in the user's manual). When you are satisfied with the configuration settings, click on the **OK** button. A telnet session window will open and you will be prompted to log in to the host you specified as the Session Name. Log in to the host as usual.

3. After you are logged in, you can further customize the session. Move the session window to the location on the desktop where you would like it to appear every time you connect to this host. Next, open the Session menu and explore the options for controlling the appearance and nature of your session. You can change the style and size of the display font, the maximum number of lines in the session window, the function of the **delete** key, the echo mode, and many other parameters. If you have a color monitor, you may want to change the foreground and background colors—also an option under the Session menu. (**Note:** the customization options available in the Configuration box and Session menu vary from one version of telnet to the next. The most recent release, *NCSA/BYU Telnet 2.5*, offers the richest set of features to date.)
4. When all aspects of the session are configured as desired, it's time to save the session information to a file. To do so, open the File menu and select the **Save Set...** option. A standard Save dialog box will appear, and you will be

prompted to enter a name for the set. After you have entered a name, click on the **OK** button and the current settings will be saved to a file with the specified name. If you use NCSA Telnet 2.4 or earlier, the icon for the configuration set file will look like the one on the left below. If you use release 2.5, the icon will resemble that on the right.



ux1



ux1

5. The next time you want to start a session using the same settings, double click on the icon of the configuration set you just saved. This will both start the Telnet software (if it is not already running) and open a telnet session with the host you specified during step 2 above.

### A Few Extra Pointers

**Saving Multiple Sessions.** A telnet configuration set can contain more than a single session window. If you are accustomed to working with multiple sessions at once (perhaps you keep both a ux1 and a uxh session open most of the time), you can save multiple session windows in a single configuration set. To create a configuration set that contains two or more sessions, open each session successively as described in steps 1 through 3 above. Once you have customized all sessions and placed the session windows exactly where you want them on the desktop, save the set as described in step 4. When you open the set by clicking on the icon, the two or more sessions saved in the set will open concurrently.

**Adding Your Configuration Set to the Apple Menu.** If your Macintosh is running System 7 or higher, you can add your favorite telnet configuration sets to the Apple Menu. To do so, move the configuration set icon (or an alias of the icon) to the **Apple Menu Items** folder in your **System** folder. Once the set file is in this folder, you can start telnet and open the set by selecting its icon/name from the Apple menu.

For help with questions and problems concerning the configuration of NCSA Telnet for the Macintosh, call the CCSO Microcomputer Consultants at 244-0608.

- Lynn Ward

# Expanded Network Access to Library Services

## CAMPUS NEWS



Are you tired of seeing the "Sorry, all lines are in use" message when trying to connect to the online library catalog? If so, there's good news ahead. CCSO and AISS are in the process of consolidating and expanding network access to library information services.

Until recently, UIUCnet users could only access online library services through one of CCSO's two TCP/IP gateways, *IO Plus* and *LCSgated*. Due to hardware constraints, these two gateways are limited to a total of twelve simultaneous connections (eight ports are used by *IO Plus* and four by *LCSgated*). AISS now offers a faster, more robust service, which supports over forty concurrent TCP/IP (telnet or tn3270) sessions and provides access to the full range of library services, including *ILLINET Online*, *IBIS*, *CARL UnCover/UnCover2*, and the University of Illinois at Chicago's online library catalog. (For more information on *ILLINET Online*, *IBIS*, and *CARL UnCover*, see the Dec. 91 - Jan. 92 issue of *UIUCnet*, vol. 4 no 5.) Since AISS is able to fully support network access to library services with its own equipment, CCSO probably will phase out its interim library gateway services. Instructions on how to use AISS facilities to connect to library services via the campus network are provided below.

### Two Modes of Access

Most of the databases that make up the online library system reside on an IBM mainframe, which is part of the Administrative Computing Network (ACN) at the University of Illinois at Chicago. This mainframe can only communicate directly with IBM 3270-class terminals or computers running software that can emulate a 3270 terminal, such as the special version of telnet called *tn3270*. Since the *tn3270* application is not available on all computers on the campus network, AISS offers two methods for connecting to the library system, one for 3270/tn3270 users and another for plain telnet users. Although it's generally preferable to access library services on the ACN with a 3270 terminal or 3270 emulation software, it's not always

easy to tell whether the *tn3270* software is on your computer. The sidebar on page 9, **How to Connect to Library Services via TN3270 and Telnet**, gives advice on how to figure out whether *tn3270* is available on a particular system and how to open a *tn3270* or *telnet* session with the library.

### Getting Started

If you have access to a TCP/IP 3270 emulation utility, you can establish a direct connection with the ACN's IBM mainframe UICMVSA by opening a *tn3270* session with the host named **uicmvsa.aiss.uiuc.edu**. Once connected, you should see the ACN welcome screen, which is sometimes called the **ABC** screen (see Figure 1 below).

```
CHICAGO (312) 996-4806-----HELP DESK-----URBANA (217) 333-3102
EVSELEN
Welcome to the UNIVERSITY OF ILLINOIS
Administrative Computing Network

A Administrative Computing Network Services (ACN)
B Library Services
C Public Services

NOTICE
Unauthorized use of University of Illinois computerized systems, data, or
resources, or an attempt to gain unauthorized access is prohibited by
University policy and may constitute a violation of Illinois state law.

To select a service, type the letter and press Enter at the Command ==>
prompt. To exit, type X and press Enter at the Command ==> prompt.

Command ==>
F3=Exit
```

Figure 1: The welcome screen of the Administrative Computing Network, often called the **ABC** screen.

```
Welcome to the ILLINET Online Network
Press b and <RETURN> to login.
```

Figure 2: The *ILLINET Online* welcome screen on **illinet.aiss.uiuc.edu**.

If a *tn3270* application is not available on your system, you can open a plain telnet session with the intermediary host **illinet.aiss.uiuc.edu**. A screen similar to that shown in Figure 2 will appear. Type a lower case **b** and press the **<enter>** key. You should then see a screen like the one shown in Figure 3. Select option 1 and this intermediary host will open a *tn3270* session with UICMVSA for you, serving as a translator between your computer and the IBM mainframe. (**Note:** Selecting option 2 will display a map of the keystrokes used

email questions to bugs@illinet.aiss.uiuc.edu

#### Selections

- 1) ILLINET Online Network Services
- 2) Keyboard map

E - Exit

Enter your selection here==>

Figure 3: The *ILLINET Online* access screen on **illinet.aiss.uiuc.edu**.

to emulate special keys found on an IBM 3270-class terminal. For more information, see the section called **Keyboard Mapping** below.) Eventually, you should see the **ABC** screen shown in Figure 1.

### Accessing Library Services

Once you are at the **ABC** screen, type the letter **b** and press the **<enter>** key to enter the Library Services region of the ACN. You should see a screen similar to that shown in Figure 4. Select the desired service by entering its number at the **Command==>** prompt. For the most part, the services labelled **IOMENU** (the full-screen *ILLINET Online* catalog), **IBIS**, and **CARL** function exactly as they do when accessed through CCSO's *IO Plus* gateway. (All three are menu-driven systems with online help. Most operations involve entering single letter commands or filling in empty fields with search criteria.) There is, however, one minor change that will affect *ILLINET Online* (**IOMENU**) users. When you enter *ILLINET Online*, you will be presented with a screen resembling that shown in Figure 5 on page 10. If you press the **<enter>** key, the default scope for searches will be all 800+ libraries in the statewide system. To limit searches to

```
Library Services
CHICAGO (312) 996-4806-----HELP DESK-----URBANA (217) 333-3102
LVSELEN
Node: TCP00013


NO SERVICE SERVICE DESCRIPTION TYPE STATUS
-----
1 CARL UnCover-Journal Contents Multi
2 IBIS Ill. Bibliographic Info System Multi
3 IOMENU Illinet On-line Catalog Multi
4 LUIS_UIC UIC-Library Catalog(Luis) Pass

To select a service, type the number and press Enter at the Command ==>
prompt, or tab to the service line and press Enter. To exit, type X and
press Enter at the Command ==> prompt.

Command ==>
F3=Exit
```

Figure 4: The Library Services menu.  
(continued on page 10)


# How to Connect to Library Services via TN3270 and Telnet

 **From a DOS Machine:** If you use public domain TCP/IP software such as NCSA or Clarkson Telnet, change to the directory containing the telnet software and list all executable files beginning with the letter "t" by typing `dir t*.exe` at the DOS prompt. Look for the files called `tn3270.exe` and `telbin.exe` in the directory listing. If the file `tn3270.exe` is present, open a session with UICMVSA by entering the command `tn3270 uicmvsa.aiss.uiuc.edu` at the DOS prompt. If you can only find `telbin.exe`, open a plain telnet session with the intermediary host by typing `telbin illinet.aiss.uiuc.edu` at the DOS prompt.

Versions 2.05 and later of the commercial TCP/IP software *PC/TCP* incorporate both telnet and tn3270 into a single application with the filename `tn.exe`. To access library services with this package, change to the directory containing the *PC/TCP* software and enter the command `tn uicmvsa.aiss.uiuc.edu` at the DOS prompt.


Some DOS computers have a `telnet.bat` file that automatically invokes `tn3270.exe` or `tn.exe` when you enter a command like `telnet uicmvsa.aiss.uiuc.edu` (for example, the DOS computers at CCSO public sites are set up in this manner). If there is a `telnet.bat` file on your computer, you might want to try this command. If it takes you to the screen shown in Figure 1 on page 8, using the `telnet.bat` file is probably the best method to connect to UICMVSA and other remote systems, even those requiring plain telnet.


If you use a different commercial or public domain TCP/IP package, or the suggestions above do not work for you, contact your building network administrator or the CCSO Microcomputer Consultants (244-0608) for help.

 **From a Macintosh:** Look for an application called `tn3270` or `tn3270-MacTCP` on your Macintosh. (System 7.x users can use the **Find** utility in the


File menu and System 6.x users can use **Find File** utility in the Apple menu to search for a file with the name `tn3270`.) If the `tn3270` application is on your Mac, start the software. A pop-up window will display the names of all `tn3270` settings files in the current folder. If you see a file called **AISS**, **ACN**, **UICMVSA**, or something similar, try opening it and it may take you exactly where you want to go. If not, click on the **Cancel** button to close the window and select the **Hostname** option under the Settings menu. Enter the hostname `uicmvsa.aiss.uiuc.edu` into the dialog box and click on the **OK** button. Then, pull down the File menu and select **Open Connection**. If all goes well, you should eventually see a screen similar to that in Figure 1 on page 8. (Note: The foregoing describes a quick and not-so-elegant method for setting up a `tn3270` session. Talk to your network administrator or read the manual to learn how to optimize the various settings for your particular hardware.)


If `tn3270` is not on your Mac, use *NCSA Telnet* to open a connection with `illinet.aiss.uiuc.edu`.

 **From VMD:** Since VMD is an IBM mainframe, the version of telnet installed on this machine provides 3270 emulation by default (in fact, it works best when communicating with other IBM mainframes). To open a direct connection with the library, first link to the disk containing the telnet software by entering the command `linkto tcpip` at the system prompt. Then, enter the command `telnet uicmvsa.aiss.uiuc.edu` to open a session with UICMVSA.

 **From a Unix Machine:** Unix machines may or may not have `tn3270` installed (most CCSO-administered machines do). The best way to find out is to try opening a connection using the command `tn3270 uicmvsa.aiss.uiuc.edu`. If you get an error message like "Command not found," the machine probably doesn't have

`tn3270`. If `tn3270` is not available, open a plain telnet session with `illinet.aiss.uiuc.edu` to access library services. (Note: Even if the `tn3270` command doesn't work on your Unix system, it's possible that the `tn3270` application is installed, but has a different name. For more information on the remote login utilities available on your system, talk to your system administrator.)

 **From a DEC/VMS Machine:** VMS machines may or may not have `tn3270` installed. The best way to find out is to try opening a connection using the command `tn3270 uicmvsa.aiss.uiuc.edu`. If you get an error message like "unrecognized command verb - check validity and spelling \TN3270\," the machine probably doesn't have `tn3270`. If `tn3270` is not available, open a plain telnet session with `illinet.aiss.uiuc.edu` to access library services. (Note: A popular set of TCP/IP tools for VMS machines called *Multinet* by TGV has a telnet utility that negotiates terminal type on the fly and switches to 3270 mode when necessary. If this software is on your system, you can use the command `telnet uicmvsa.aiss.uiuc.edu` to access the ACN directly. For more information on the remote login utilities available on your VMS system, talk to your system administrator.)

 **Dialing in from a Computer with a Modem:** Although it is possible to conduct a telnet or `tn3270` session from the CCSO terminal servers, modem users are urged to use the dial-up facilities provided by the library. Dialing either 333-2494 or 333-8269 will connect you to a pool of modems maintained by the library. The brochure called *Remote Access to ILLINET Online and IO Plus (IO+)*, available at the main library and many departmental libraries, gives detailed information on how to use the library's dial-up facilities. ■



## Library Services...

(continued from page 8)

items at the UIUC campus, press the **<tab>** key to move to the **scope code** field and enter **uc**; or enter **o** at the **==>** prompt to see a complete list of the libraries you can search.

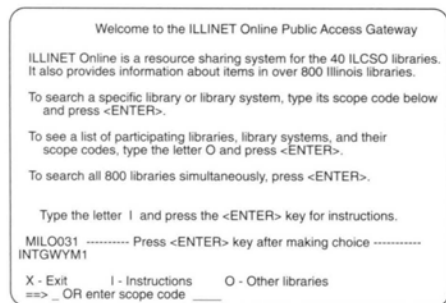


Figure 5: The ILLINET Online welcome screen. Press **<enter>** to search all 800+ libraries or press **<tab>** and enter **uc** to search the UIUC Library.

Option 4 (LUIS\_UIC) in the Library Services menu provides access to the library catalog of the University of Illinois at Chicago. Like the other three services, LUIS is an intuitive, menu-driven system. However, getting on and off the LUIS system is a little tricky. When you select option 4, a screen of instructions will appear briefly and then the word **Port** followed by a number will appear in the upper left corner of an otherwise blank screen. To get past this screen, you must enter your keyboard equivalent for the 3270 **clear** function. To exit the LUIS system, enter your keyboard equivalent for the 3270 **PF3** key. (See the next section for information about 3270 keyboard mapping.)

## Keyboard Mapping

IBM mainframe applications such as the various library services offered on the ACN often make use of special keys that are typically found on the IBM 3270 family of terminals. If you are using 3270 emulation software, normally there are alternative keystroke sequences that can be substituted for 3270-specific keys. The good news is that the online library applications only make use of three such keys: **PF3**, **clear**, and **reset** (use **reset** only if your session locks up and you can't enter any new commands). The bad news is that there is no universal standard for keyboard mapping. Thus, the keystrokes re-

quired to send the equivalent of the **PF3**, **clear**, and **reset** keys vary from one version of tn3270 to the next. Here are some guidelines to help you figure out which keystrokes to use:

- Most 3270-specific keys are mapped to keystroke sequences that use the **<esc>** key or the **<ctrl>** key in conjunction with one or more ASCII characters. For key sequences using the **<esc>** key, press and release the **<esc>** key before entering any additional keystrokes. However, when entering a sequence that uses the **<ctrl>** key, depress the **<ctrl>** key and do not release it until you have pressed the remaining keystrokes required for the action.
- Although there are no fixed rules for 3270 key mapping, there are a few common conventions that you can try if you don't have ready access to the keyboard map for your tn3270 software. The **PF** keys (programmable function keys, e.g., **PF1**, **PF2**, etc.) usually use the sequence **<esc> #**, where # is the number following **PF**. Thus, pressing **<esc> 1** is often equivalent to **PF1**, **<esc> 2** to **PF2**, and so forth, up through **PF9**. **PF10** through **PF12** continue to use the top row of alpha-numeric keys: **<esc> 0 = PF10**, **<esc> - = PF11**, **<esc> = = PF12**. For **PF** keys higher than **PF12**, begin with the top row of keys again, this time with the **<shift>** key; for example **<esc> ! = PF13**, **<esc> @ = PF14**, **<esc> # = PF15** and so forth. The key sequences **<ctrl> z** and **<ctrl> l** (the lowercase letter "l," not the number 1) are commonly used for the 3270 **<clear>** key.
- If you access library services with plain telnet at the address **illinet.aiss.uiuc.edu**, the special 3270 keys follow the conventions for mapping **PF** keys described above. The key combination **<ctrl> l** is used for the **<clear>** key and **<ctrl> r** is used for **<reset>**. A complete keyboard map can be displayed when you log on.
- If you access library services at the address **uicmvsa.aiss.uiuc.edu**, the keystrokes you use will depend on how your specific tn3270 application maps your keyboard. The AISS publication *TCP/IP—Use and Instructions*

gives the default keyboard maps for PC/TCP's telnet/tn3270 and Clarkson's tn3270 applications for DOS computers and Brown's tn3270 for the Macintosh. This publication can be requested by calling the AISS Help Desk at 333-3102. Typically, tn3270 applications for the Mac and PC use the function keys on the PC and extended Mac keyboards as substitutes for the 3270 **PF** keys. Thus, **PF1=F1**, **PF2=F2**, and so on. Consult your tn3270 software manual for more specific information about keyboard mapping.

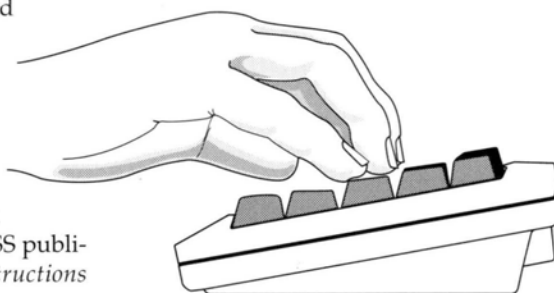
On Unix machines, the tn3270 keyboard map is usually an ASCII file called **map3270** located in the **/etc** or **/usr/local/etc** directory. A set of comments at the beginning of the file explains how it works.

Users who access VMD through the 7171 controller (with a dumb terminal or a computer with terminal emulation software) can view VMD's 3270 keyboard maps by entering the command **help 7171 termtype** (e.g., **help 7171 vt100**) where *termtype* is the standard abbreviation for your terminal type—i.e., the same name you use when you respond to the **Enter Terminal Type:** prompt before you log on.

## For More Information

Report problems or bugs encountered on **illinet.aiss.uiuc.edu** by sending an e-mail message to **bugs@illinet.aiss.uiuc.edu**. If you need help logging on to the library system, call the UIUC Library Telephone Center at 333-8400 or CCSO Systems Consulting at 333-6133. Finally if you have questions about a particular library application such as IBIS or ILLINET Online, contact the UIUC Library Information Desk at 333-2290. ■

- Lynn Ward



# UIUCnet E-mail to Fax Service Improved

## CAMPUS NEWS



The old CCSO e-mail to fax gateway has been replaced with newer, faster hardware (two fax modems running on ux2, an IBM RISC 6000/540) and more flexible software. In the short term, users won't notice much difference in the way the gateway works, but there are plans to implement some exciting new features.

### What is the E-mail to Fax Gateway?

If you are new to the campus network, you may be unfamiliar with the e-mail to fax gateway. In a nutshell, this convenient service allows you to send an e-mail message to any fax machine in the United States and Canada. The procedure is quite simple. When preparing your e-mail message, enter the following information in the **To:** or **cc:** field of the header of the message:

`name_phone#@fax.uiuc.edu`

Replace the variable *name* with the name of the person to whom you are sending the fax and replace the variable *phone#* with the phone number of the destination fax machine. For the *name* section of the address, you may use the letters A-Z and a-z, the numbers 0-9, and the hyphen (-) and underscore (\_) characters. However, the *name* section must not end with a digit. When entering the phone number, use only the numbers 0-9 with no punctuation marks. Off-campus fax numbers need not include the "9" prefix, which is normally used to get an outside line. Local off-campus numbers should **not** include the "217" area code. Also, the "1" prefix is only required for long-distance calls that are within the 217 area code (e.g., 12174420001). There are no limits on the length of the text that precedes the "@fax.uiuc.edu" part of the address, so it's possible to include the name and office address or other information about the recipient. For example:

`Joe_Kaufmann_PC_Wholesalers_7089221111@fax.uiuc.edu`

Once you have finished preparing your message, send it as you would any other e-mail message. The message will be routed

to the fax gateway, and the ASCII text of the message will be converted to a bitmapped image that can be printed out on the receiving fax machine. If you would like further details on how to use the e-mail to fax gateway, pick up a copy of *CCSO User Guide #301, CCSO E-mail to Fax Gateway* at the CCSO Resource Center, 1420 DCL.



### Facts about the New Fax Gateway

The quality of the output and general performance of the new fax gateway significantly surpasses its predecessor. Senders and recipients are likely to notice the following changes and improvements:

- A handsome cover page accompanies every outgoing fax (it is no longer possible to suppress the cover page as with the previous fax gateway). Included on the cover sheet are a University banner (with address and telephone number), the name of the recipient, the phone number of the receiving fax machine, the e-mail address of the sender, the date, the subject of the fax (if the sender fills in the **Subject:** field of the e-mail header), and the total number of pages including the cover. In addition, a header with a time/date stamp and e-mail address of the sender appears at the top of every page.
- The resulting fax image is noticeably sharper than that produced by the old gateway. Unless special codes are embedded in the message (see the **Future Enhancements** below), the default font is 10 point Courier at 12 characters per inch.
- Message line lengths are now limited to 80 characters per line. Lines longer than 80 characters will be truncated.
- If the first attempt to send the fax is unsuccessful because the receiving fax

machine is busy, the gateway will try to send it six more times at increasing intervals within a 90 minute window. When compared to the four retry / eight minute window of the old fax gateway, the new retry algorithm greatly increases the chances for successful transmission. As in the past, the sender will be notified via e-mail about the success or failure of the transmission.

### Future Enhancements

The fax gateway software supports PostScript files, files in the Hewlett Packard PCL Level IV language (the printer command language used by HP LaserJet Series II printers), and several other file formats. Unfortunately, not all e-mail packages are able to process such files correctly, and under some circumstances, when PostScript or PCL files are incorporated into an e-mail message, the resulting fax can be tens or hundreds of pages of raw printer instructions. A utility for sending PostScript, PCL, TIFF and other file types to the fax gateway is currently under development. Once this utility is in place, it will be possible to send faxes that include graphic images and standard Adobe PostScript fonts.

Faxes are presently restricted to destinations in the U.S. and Canada, but CCSO does plan to offer an international e-mail to fax service. While domestic faxes are currently free, there will be a charge for international faxes. More specific information about international faxes and support for alternative file formats will be announced when these services become available.

### Restrictions and Support

The CCSO e-mail to fax gateway will only accept messages from users in the **uiuc.edu** and **uic.edu** domains. The fax gateway is available to University faculty, staff, and students, but is intended for University-related business only (so don't use it to order a pizza!). Abuse of the gateway could result in the loss of CCSO services for the offender.

If you encounter problems with the new gateway, send an e-mail message to `fax-trouble@ux1.cso.uiuc.edu`. CCSO Systems Consultants (333-6133) can also answer basic questions about the service. ■

- Lynn Ward



1120 DCL  
Campus MC 256  
1304 W. Springfield Ave.  
Urbana, IL 61801

GIFT & EXCHANGE DIV  
MAIN LIBRARY  
ROOM 314  
CAMPUS  
MC-522

## Eudora 1.3...

(continued from page 4)

### Attention All Modem Users!

As you may recall, CCSO is phasing in a plan that will require all dial-in users to log in to the terminal servers before they can access CCSO machines and other networked computers on and off campus (for more information, see the October 1992 issue of *UIUCnet*, vol 5 no. 6). Terminal server login will be required to access CCSO machines as early as May 18, 1993. If you use Eudora with a modem, it is critical that you upgrade to release 1.3 before terminal server login is enforced. The latest release of Eudora has a **Dial-up username:** field in the configuration box. Enter your ph alias into this field. You also will need to get the file called **CiscoNavs** and place it in the appropriate location in your **System** folder (see the section called **Upgrading to Eudora 1.3** below).

Once you have installed Eudora 1.3, you will be prompted for two passwords when you send or check for mail. Enter your ph password at the first password prompt and your login password at the second prompt.

### Obtaining Eudora 1.3

Eudora 1.3 is available via anonymous ftp from <ftp.cso.uiuc.edu> in the **mac/eudora** directory. Be sure to grab **StuffIt-Expander.hqx** because some of the archived files in the Eudora distribution are stubborn and can only be "unstuffed" with this application.

Eudora 1.3 can also be obtained at the CCSO Resource Center, 1420 DCL. It is available on the AppleShare volume called **Public**, which is mounted on every Mac in

the center. Eudora 1.3 is in the **Software/Mac/Communications/Eudora 1.3** folder. The application itself and the **CiscoNavs** file (for modem users) are in the **Program** folder. Various pieces of documentation including the full 1.3 user's guide (in *PageMaker* 4.2 format), release notes, a HyperCard stack of commonly asked questions and answers, etc. are located in the **Documentation** folder. If you are just upgrading from an earlier version of Eudora, you shouldn't need anything else. But, if you are installing Eudora for the first time, ask a Resource Center consultant to help you figure out exactly which additional files you'll need. A printed version of the Eudora manual can be purchased at the Accounting and Distribution desk in the Resource Center. The price is \$6.00 and covers the cost of duplication and binding.

### Upgrading to Eudora 1.3

Upgrading to Eudora 1.3 is a snap. If your Mac is attached to the campus network, just replace your old Eudora application file with the new application file. Eudora 1.3 will look for existing configuration and mailbox information in the **Eudora** folder inside your **System** folder. All your previous Eudora settings will be used unless you decide to change them.

If you use Eudora with a modem, you also must install the **CiscoNavs** file. System 7.x users should place the **CiscoNavs** file in the **Preferences** folder inside the **System** folder. (Do not just drag the **CiscoNavs** file onto the **System** folder. Unlike many System 7 preference files, it will not be placed in the **Preferences** folder automatically.) System 6.x users should copy the file to the top level of the **System** folder.

### For More Information

For more information on obtaining or installing Eudora 1.3, contact the CCSO Microcomputer Consultants at 244-0608. Bug reports should be directed to Steven Dorner via e-mail at [sdorner@qualcomm.com](mailto:sdorner@qualcomm.com).

■ - Lynn Ward

### About UIUCnet

*UIUCnet* provides timely information about the campus network. It is published by the Computing Services Office and is edited by Lynn Ward. Comments and suggestions for topics are welcome. Permission to reprint all or part of *UIUCnet* for non-profit purposes is granted, provided full acknowledgement of the source is given. Feel free to reach us via electronic mail ([uiucnet@uiuc.edu](mailto:uiucnet@uiuc.edu)), campus or U.S. mail (UIUCnet, Computing Services Office, 1120 Digital Computer Laboratory, MC 256, 1304 W. Springfield Ave., Urbana, IL 61801), or by phone at (217) 244-0681.

For a free subscription, just send us your name and (preferably campus) address; be sure to include your department and mail code if you send a campus address. *UIUCnet* is also available in electronic form. Recent issues are posted to the Netnews newsgroup **uiuc.pubs.uiucnet**. Back issues are available for download from the anonymous ftp host <ftp.cso.uiuc.edu> in the directory **doc/net/uiucnet** and from the UIUC Gopher server.



# UIUCnet

The University of Illinois Campus Network

Vol. 6 No. 3

April-June 1993

## Icon Key

THE LIBRARY OF THE

### Audience



Novices



Experienced users



Network Administrators

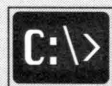


AISS clients



Everyone

### Platform/Operating System



PC compatibles (DOS™)



Macintosh™



X Window System™



NeXT™



CCSO'S IBM™ mainframe, VMD



UNIX™ mainframes and workstations



Any platform

JUL 2 1993

UNIVERSITY OF ILLINOIS  
 URBANA-CHAMPAIGN

## PCs are POPping with PC Eudora and NUPop

### CLIENTS FOR CLIENTS

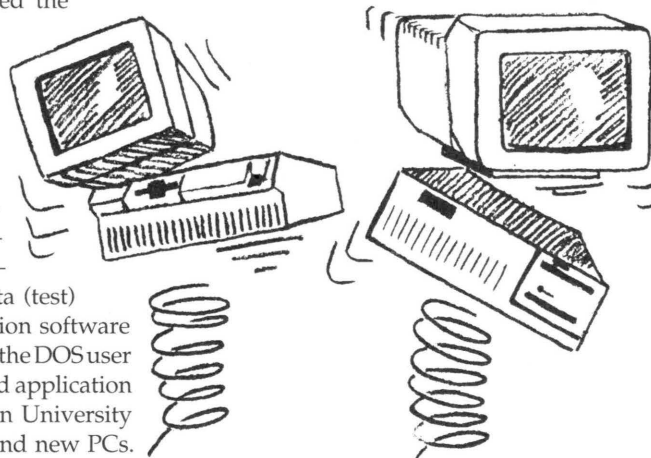


When Eudora, the well-known e-mail client for the Macintosh, was released at UIUC a little over two years ago, it literally revolutionized the way Mac users on the campus network (and eventually all over the world) processed their electronic mail. Eudora afforded people the luxury of preparing outgoing messages and reading and organizing incoming messages within the familiar confines of the graphical, menu-driven Macintosh desktop. For most Mac users—even those well acquainted with UNIX e-mail software and text editors—it was time to say, "Good-bye Elm and vi, hello Eudora."

Shortly after Eudora was introduced, PC users began clamoring for a similar e-mail package for DOS machines. Several DOS-based e-mail clients entered the public domain, but until recently, most were either unstable, unsupported, or lacking the extraordinary functionality and ease-of-use that made Eudora an overnight sensation. However, two packages—NUPop and PC Eudora—previously available in beta (test) versions, are now production software and hold much promise for the DOS user base. NUPop is a text-based application developed at Northwestern University that can run on both old and new PCs. For high-end PCs and compatibles,

QUALCOMM, Inc. (current developer of Mac and PC Eudora) offers a Microsoft Windows version of Eudora that looks and feels very much like its Macintosh cousin.

Like Eudora for the Mac, both NUPop and PC Eudora are POP (Post Office Protocol) clients. In order to use a POP client, you must have an e-mail account on a multi-user computer that is running POP server software (all CCSO-administered mainframes including uxa, uxh, ux1, ux4 and VMD support the POP protocol). Most of the actions you will perform with the client (NUPop or PC Eudora) do not involve network communications. For example, when you prepare a new message, you use a text-processor built in to the POP client. Reading, organizing, and replying to incoming mail are also desktop operations. The only time the network comes into play is when you send or retrieve mail (or use special network utilities such as ph, finger, etc.).



(continued on page 2)

## PC Eudora and NUPop...

(continued from page 1)

PC Eudora and NUPop both possess a rich set of features and configuration options comparable to those found in Eudora for the Mac. Some of the more outstanding of these are:

- **Integrated Text Processor.** Creating new messages or replies is a breeze with the intuitive word processing capabilities built in to NUPop and PC Eudora. Word wrap and the ability to select, cut, copy, and paste blocks of text are standard features. Both packages also allow the user to open existing text files in order to paste all or part of the file into a new message.
- **Nicknames/Groups.** You only need to type an Internet or BITNET address once to realize that such addresses are often long and difficult to remember. With PC Eudora and NUPop you can create aliases (called *Nicknames* in Eudora and *Groups* in NUPop) for the people or groups with whom you correspond regularly. Creating an alias involves entering a user's full e-mail address and a short, easy-to-remember nickname that corresponds to the full address. For example, an alias for the address of a colleague with the e-mail address `mgeg8538@uxa.cso.uiuc.edu` might be "Mel." Once an alias is created, you can use it in the **To:** or **cc:** field of the message instead of the full e-mail address. The client software takes care of expanding the alias to the full address when it sends the message to the recipient.
- Nicknames or groups can also consist of multiple e-mail addresses or even other nicknames. For example, the nickname "happy hour" might include the full e-mail address (or alias) for each member of the gang you meet after work on Friday afternoons. **Helpful Hint:** You can use the copy and paste feature to copy an e-mail address from the header of a message or the ph query window into your list of nicknames or groups.
- **Binary and ASCII File Attachments.** One of the shortcomings of electronic mail is that many systems can only process plain ASCII text, and thus it is often not possible to send binary data such as

formatted word-processing files, spreadsheets, executable programs, etc., by e-mail. NUPop and PC Eudora get around this problem by encoding binary files in the format known as BinHex (for more information about BinHex format, see the November 1992 issue of *UIUCnet*, vol. 5 no. 7). The binary data is converted to a format that uses only ASCII characters and is attached to the body of the e-mail message. When a BinHexed attachment is received by PC Eudora or NUPop, the attachment is automatically restored to its original state and placed in a directory designated by the recipient. If the recipient uses an e-mail package that cannot decode BinHexed files, a special utility can be used to convert the file back to its native format.

ASCII text files can be attached to any NUPop or PC Eudora message and do not require special handling. They simply appear as part of the body of the message.

- **Mailboxes and Folders.** If you receive lots of e-mail, it's very helpful to be able to store it according to your own organizational scheme. This need is addressed in PC Eudora and NUPop with the Mailbox menu. Both packages come preconfigured with a few essential mailboxes such as **IN**, **OUT**, and **Trash**. New mailboxes can be created by selecting the appropriate command from the Mailbox menu. Once a new mailbox has been created, messages can be moved from one mailbox to another. When creating a mailbox in PC Eudora, you have the option of designating it as a folder, which is a mailbox that contains other mailboxes rather than messages. Mailboxes can thus be neatly organized into categories that form a hierarchical filing system.
- **Automatic Message Retrieval.** NUPop and PC Eudora can be configured to check for and download new mail from a POP account at regular time intervals. A pop-up window and/or alarm signals the arrival of new incoming messages. (**Note:** because NUPop is a DOS application, it is only convenient to take advantage of timed message acquisition when running NUPop as a background process under multitasking software such as MS Windows.)

- **Built-in Ph Client.** Standard queries can be sent to the CCSO Nameserver database with the built-in ph client found under the Special menu in PC Eudora or the Utilities menu in NUPop. Ph makes it easy to look up e-mail addresses, phone numbers, and many other types of other information about people and units on campus. You can also use it to get weather, area code, campus timetable, and local restaurant information. Also included in NUPop are a finger client (for looking up information about a user account on a specific host), the *IP Finder* (an *nslookup* client, which will return the numeric IP address associated with any fully-qualified domain name), a simple telnet client for logging in to remote hosts, and a *webster* client for looking up information on Webster Dictionary servers.

## System Requirements

If you decide to run a POP client on your PC, which application should you use, PC Eudora or NUPop? Your choice will be partially influenced (if not dictated) by your PC hardware and software. The system requirements for PC Eudora are not met by many PCs on the campus network. For optimal performance, PC Eudora requires a 386 processor or better, at least 2 MB of RAM, a sizable hard disk (PC Eudora only requires 750 KB, but MS Windows uses roughly 10 MB without any other applications installed), mouse, and a color VGA or better resolution video subsystem (PC Eudora can run on a 286 machine with a Hercules monochrome graphics or EGA video card, but performance will suffer). On the software side, two commercial products—Microsoft Windows version 3.1 and FTP Software's *PC/TCP* release 2.1 or higher—must be installed on your machine in order to run PC Eudora. (For information about the UIUC site-license for PC/TCP, contact the CCSO Accounting and Distribution Desk, 1420 DCL, 333-7752).

NUPop, on the other hand, can run on both low- and high-end machines. The pseudo-graphical user interface (consisting of windows, buttons with drop-shadows, and pull-down menus) uses only characters from the ASCII and IBM extended ASCII character sets. Thus, even a plain monochrome video card and monitor will suffice. The NUPop executable and associated files take up roughly 700 KB of hard

(continued on page 3)

## PC Eudora and NUPop...

(continued from page 2)

disk space and require 470 KB of free RAM after DOS and memory resident programs are loaded. The only special software required by NUPop is DOS version 3.0 or higher and a *packet driver* for your network interface card. (Packet drivers for common brands and models of network cards are free and ftp'able from many locations on the Internet. In any case, most PCs on UIUCnet already have a packet driver installed. For more information on packet drivers, see the **NetWord** and **Administrivia** columns in this issue.) Power users can run NUPop in a DOS window as a background operation under MS Windows. In short, just about any PC with a hard disk and at least 640 KB of RAM can run NUPop.

Apart from system requirements, there are a few other items to consider before committing to NUPop or PC Eudora. Although the two programs are roughly equal in terms of overall functionality, each possesses its own unique set of features and flaws.

### More NUPop Facts and Features

NUPop supports both plain serial and SLIP (Serial Line IP) connections by way of the CCSO terminal servers, a feature not yet available in PC Eudora. This means you can run NUPop from home or office with just a PC and modem (no direct network connection is required). If configured for serial communication, NUPop automatically dials the terminal server and connects to your mail server when you issue the command to send or retrieve mail. Once the transfer is complete, the connection is closed. Special NUPop script files customized for the CCSO terminal servers are available with the NUPop distribution files at the CCSO Resource Center.

Although both NUPop and PC Eudora are much easier to use with a mouse, most NUPop commands can be easily managed from the keyboard (running PC Eudora without a mouse, on the other hand, is extremely cumbersome). Menu and button commands are executed by pressing the <Alt> key in conjunction with the highlighted letter in the command name itself. For example, to open the **Options** menu, use the key combination <Alt - O> (see Figure 1). The space bar acts as a toggle for selecting and unselecting messages within

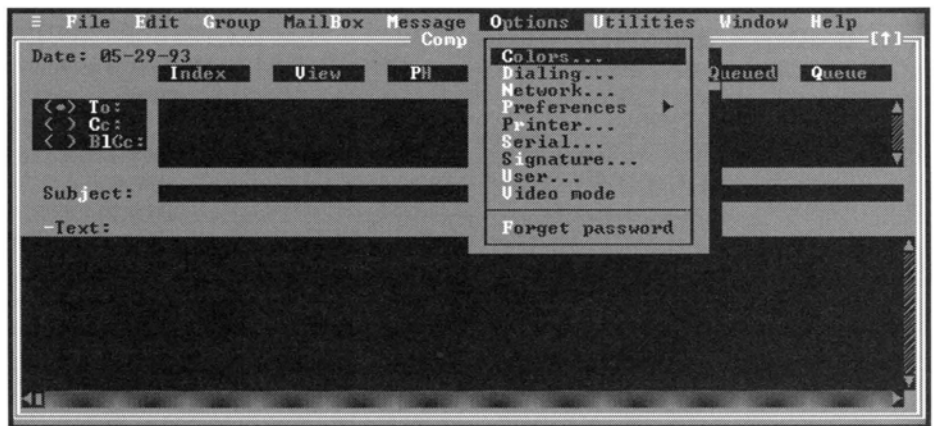


Figure 1: NUPop commands can be executed and menus can be opened with a mouse or by pressing the <Alt> key in combination with the highlighted letter; e.g., <Alt-O> for Options. The following files are available:

**nupopx.zip** - ZIP archive of NUPop (where the variable *x* stands for the current version number [**nupop103.zip** as of this writing])

**nupopps.zip** - ZIP archive of the PostScript documentation (only useful if you have a PostScript printer)

**connect.scr** - Dialing script to connect to the UIUC terminal server with standard serial access

**slipdial.scr** - Dialing script to connect to the UIUC terminal server with SLIP access (only necessary if NUPop is the only SLIP application you use)

Both the application and documentation are in ZIP archived format. If you do not already have a utility to unzip the files, a free file-extraction utility called UNZIP is available on the anonymous ftp host **ftp.acns.nwu.edu** in the directory **pub/nupop** (retrieve the file named **unzip.exe**). When extracting the ZIP archives, you may be asked whether you want to overwrite the files named **connect.scr** and **slipdial.scr**. Enter "n" for no so that the custom UIUC script files will not be overwritten with the files for modem users at Northwestern University.

The authoritative anonymous ftp source for NUPop is **ftp.acns.nwu.edu**. Northwestern University maintains the most up-to-date versions of the software and documentation, including test versions of new releases. The NUPop application and related files are located in the **/pub/nupop**

(continued on page 4)

### NUPop Flaws

Few public domain software applications are perfect and NUPop is no exception. Until you get used to it, mastering NUPop's multiple windows is quite confusing. At start-up, the Index (IN mailbox), Composer (editor for creating new messages or replies), and Viewer (used for viewing received messages) windows appear successively on top of one another, and it is not quite clear what is happening until the action stops. Invoking the ph and/or finger utilities causes additional full-screen windows to open up. Some NUPop windows have scroll bars and others have a close box. There is actually some method behind this window madness, but, at first, it is not obvious to the user.

In general, NUPop 1.0.3 is a fairly stable piece of software, but there are still a few bugs to be ironed out. For example, occasionally NUPop will behave in an unpredictable manner or lock-up completely. Such problems do not typically result in the loss of any data but are certainly alarming when they do occur.

### How to Get NUPop

There are several ways to retrieve the NUPop software. CCSO staff have prepared a special distribution of NUPop that includes custom scripts for use with a modem and the CCSO Terminal Servers. This distribution is available on the Novell file server volume mounted on the PCs at the CCSO Resource Center, 1420 DCL (Resource Center staff can help you copy the appropriate files onto your own disks).



## PC Eudora and NUPop...

(continued from page 3)

directory and subdirectories. You may download and use these files, but CCSO staff will not offer assistance on test versions of the software.

### Installation and Support

The NUPop installation procedure may be difficult for network novices. However, an excellent installation manual and draft user's guide are available in PostScript format from the CCSO Resource Center and come highly recommended. If you do not have access to a PostScript printer, you can buy a spiral-bound, printed copy of the documentation for \$5.00 from the CCSO Accounting and Distribution Desk, 1420 DCL. This manual should answer most of your questions about installing and configuring NUPop. If you need further assistance, try contacting your building or departmental network administrator. CCSO can also offer assistance with the proper configuration and operation of NUPop. Send requests for NUPop support by e-mail to [nupop@uiuc.edu](mailto:nupop@uiuc.edu). Bug reports and suggestions should be sent by e-mail directly to NUPop developer Philip R. Burns at [pib@nwu.edu](mailto:pib@nwu.edu).

### More PC Eudora Facts and Features

Although the system requirements of PC Eudora are steep, the learning curve is not. Many of the commands in the File and Edit menus (as well as the mouse movements and keystrokes for managing individual windows) are identical to those in other MS Windows applications. The menus and commands unique to the Eudora application are largely self-explanatory and more or less identical to the Mac version.

Although PC and Mac Eudora are very similar, the Windows version offers a convenient feature not yet available on the Mac. An icon bar appears across the top of every open mailbox window, providing quick access to message management commands such as Reply, Reply All, Forward, Redirect, Print, and Trash. Once one or more messages are selected from the mailbox, clicking on the appropriate icon will produce the desired result (see Figure 2).

Finally, PC Eudora takes full advantage of the Windows multitasking environment. As a background application, PC Eudora can check for mail regularly without any direct user intervention. For users who are heavily dependent on e-mail and want a package that will notify them as soon as new mail has arrived, PC Eudora is a good pick.

### PC Eudora Flaws

PC Eudora is also not without shortcomings. Several options that appear in various menus have not yet been implemented, notably the Undo/Cut/Copy/Paste commands under the Edit menu (the standard Windows *shortcut* keys for these functions do work) and some of the switches under the Special menu. Also, at present, PC Eudora offers no support for modem users.

One of the most disconcerting aspects of PC Eudora is that it does not conform to the Windows *Multiple Document Interface* (MDI) standard. Every Eudora entity (the menu bar, each open message and mailbox, the configuration menu, etc.) exists in an independent window and appears in the Windows Task List as a separate task. If you open a message, for example, the opened message window may hide Eudora's main menu or the icon bar on the Mailbox window. Thus, it's often necessary to bring the application or a Mailbox window into the foreground in order to reply to or print an open message. In MDI-compliant applications, documents and other items generated by an application are all contained within a single window and constitute a single task. It is impossible to cover up the menu bar with a document created by the application. According to PC Eudora developers, this problem, which was a limitation of the software tool kit used to build PC Eudora, will be addressed in a future release.

### How to Get PC Eudora

PC Eudora can be downloaded from the anonymous ftp server <ftp.cso.uiuc.edu>. The application and related files are found in the `pc/pc-eudora/windows` directory. The PC Eudora distribution includes the following files:

**pcex.exe** - Self-extracting archive of the PC Eudora application (where the variable *x* stands for the current version number [**pcex10.exe** as of this writing])

**README.TXT** - A file containing basic information about the PC Eudora distribution files (if you download this file, it's a good idea to rename it because the archived application file [**pcex.exe**] also includes a file with the name **readme.txt**)

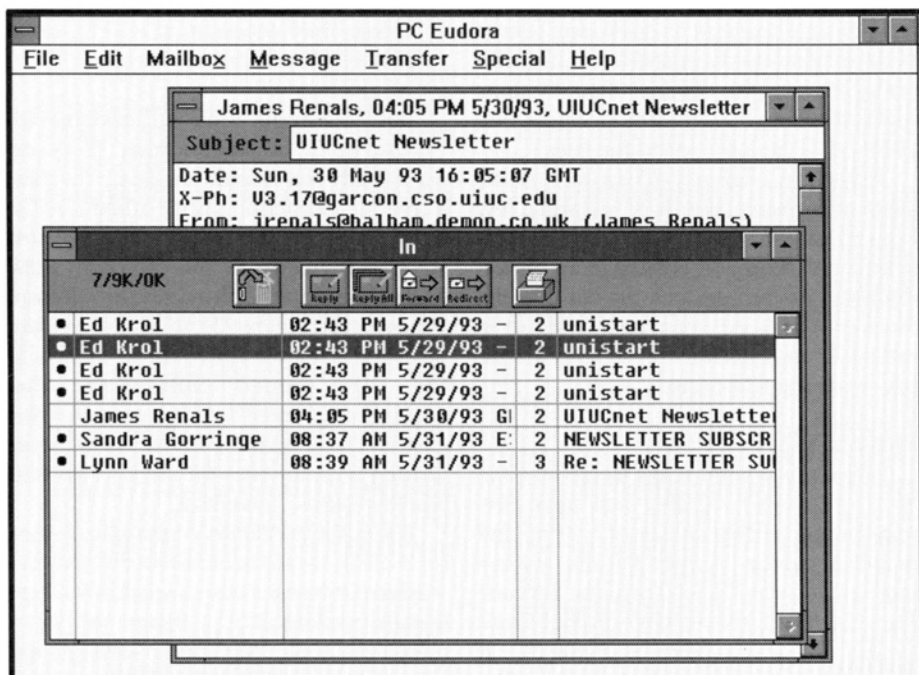


Figure 2: An icon bar appears along the top of every open mailbox window in PC Eudora.

(continued on page 5)

## PC Eudora and NUPop...

(continued from page 4)

**ftpvar.lst** - A list of FTP Software Value Added Retailers located outside the US

**wsocket.dll** - An MS Windows dynamic link library required to run PC Eudora (this file is included with PC/TCP 2.11)

If you have PC/TCP version 2.11 or later, the only file you should need is **pcex.exe**. Place this file in its own directory (e.g., C:\EUDORA) and decompress it by typing its full name. You should see an executable file named **pcapp.exe** and a text file called **readme.txt**. The **readme.txt** file gives further information for installation.

The authoritative ftp source for PC Eudora is **ftp.qualcomm.com**. The most recent production and test versions of PC Eudora are found in the **pseudora** direc-

tory on this host. CCSO will not provide help on alpha or beta versions of the software.

### Installation and Support

Instructions for installing PC Eudora are given in the **readme.txt** file included in the self-extracting archive **pcex.exe**. As with NUPop, installing and configuring PC Eudora may be too difficult for beginners, especially if PC/TCP has not yet been installed. If you do not have experience modifying your **autoexec.bat** file, creating directories and plain text files, and installing new Windows applications manually, seek out the help of an experienced user or your network administrator. The **readme.txt** file describes the information that should be entered into the configuration window under PC Eudora's Special menu once the software has been installed.

Aside from the **readme.txt** file, there is, as of yet, no documentation written specifically for PC Eudora. However, PC

Eudora and Eudora for the Mac are so similar that the Mac manual should answer most PC Eudora questions (except those regarding installation). The manual is available in PostScript format from the ftp host **ftp.qualcomm.com** (change to the **pseudora/windows** directory and download the self-extracting archive called **1\_3EUMAN.EXE**). Spiral bound copies of the Mac Eudora user's guide are sold at the CCSO Accounting and Distribution Desk, 1420 DCL, at \$6.00 per copy.

Although CCSO is not yet officially supporting PC Eudora, staff members will do their best to help users with the installation and operation of the software. Send e-mail requests for help to **eudora@uiuc.edu**, and someone will get back to you as quickly as possible. Bug reports and comments or suggestions about the software should be sent directly to the PC Eudora developers at the e-mail address **pc-eudora-bugs@qualcomm.com**. ■

- Lynn Ward

## NetWord: Packet Driver

### NETWORK



A packet driver is a piece of software that provides an interface between a network application (e.g., NCSA Telnet, Novell NetWare, LAN Manager, etc.) and a PC network interface card, usually an Ethernet card.

In the olden days before packet drivers (BPD), a network application typically talked to the interface card directly and took control of the card. This meant that the card could only support one network protocol (e.g., TCP/IP, IPX, XNS, etc.) at a time. For example, if you had loaded the software necessary to access your Novell LAN server, the card would have only been capable of sending and receiving Novell (IPX) packets. In order to use a TCP/IP package on the same machine, you would typically have had to reboot and load the TCP/IP software that talked directly to the card, at which point you would no longer be able to access your Novell server.

Another BPD problem was that each network application developer had to provide software support for many different brands and models of network interface cards. This meant updating and developing new software to support every new card that came on the market.

So, before packet drivers there were two problems: 1) a network interface card could only support one network application at a time, and 2) network application developers had to continually provide software updates to support new network cards. Packet drivers provide a solution to both problems. When a packet driver is loaded, it serves as an intermediary between a network application and the network interface card. The network application talks to the packet driver and the packet driver talks to the card. This means that the network application no longer needs to know how to communicate with hundreds of different network interface cards; it only needs to know how to talk to a piece of software that conforms to the *packet driver interface specification*. Not all network applications know how to communicate with packet drivers, but if they do, they no longer need to know how to

communicate with each and every network interface card.

Packet drivers also allow different network applications to identify themselves to the driver and keep track of each application. More than one application can talk to the packet driver at a time, so it is possible to run multiple network applications (as long as they use different protocol stacks) without any single one taking control of the network card. This means you can potentially access a Novell or LAN Manager server and run TCP/IP applications concurrently.

Most public domain TCP/IP-based applications for the PC use packets drivers. Also, a special version of IPX (Novell's native network transport protocol) developed at Brigham Young University supports the packet driver interface. There are also some special programs that allow packet drivers to coexist with Microsoft Windows and two other well-known open network interface specifications, *NDIS* (a standard developed by Microsoft and 3Com) and *ODI* (a standard developed by Novell and Apple). For more detailed information about packet drivers, see the **Administrivia** column on page 8. ■

- Lynn Ward

# Plugging in to Electronic Journals

## INTERNET TREASURES



**P**ublish or perish." This all-too-familiar maxim has controlled the lives of many young academics as they make their way down the tenure track, resulting in an astonishing proliferation of both general and increasingly specialized professional journals. To a certain extent—by making all phases of publication faster, cheaper, and easier—computer technology has contributed to this flood of scholarly output. Widespread availability of word processing and desktop publishing hardware and software, for example, has significantly reduced the time and expense involved in editing and printing academic journals. Most editors today require authors to submit a typescript and electronic copy of any article accepted for publication. Editorial changes can be made directly in the electronic version, which is ultimately used to produce camera-ready copy in house or passed on to a professional typesetter. Working with electronic text eliminates the costly, labor-intensive step of rekeying and manually typesetting an article. It also reduces the possibility of introducing new errors into the text, especially when inputting complex equations, lengthy passages in a foreign language, or other technical material.

The recent infusion of computer networking technology into the international

phers to describe serial publications whose primary method of dissemination is electronic text as opposed to the printed page. Broadly speaking, the genre includes everything from counter-culture magazines (often called *Zines*) and newsletters to peer-reviewed academic journals. Today, the several hundred electronic journals available on the Internet are dominated by less formal publications dealing with science, technology, and computer communications. However, as scholars become increasingly aware of the economic and practical advantages of electronic publication, all of academia will eventually participate in the flowering of this new medium.

### EJournal Pros and Cons

So, what exactly do electronic journals have to offer over and above printed journals? First, ejournals are extremely inexpensive to produce, distribute, and store—no paper, postage, or shelf space is required. A single issue may use as little as 15 to 150 KB of computer storage space, and not a single tree has to die on its behalf. Because ejournals are so economical, there are usually no subscription fees. Typically, one subscribes to and receives ejournals by electronic mail. Ejournals can also be produced far more rapidly than printed journals, thus delivering important or time-critical information in a timely fashion. Perhaps most significantly, electronic journal articles can be indexed and searched with tools as simple as word processing software or as sophisticated as full-text retrieval systems using multiple thesauruses, soundex routines, and fuzzy logic. When it comes to plowing through volumes of text, the human hand, eye, and brain will never be able to compete with even a run-of-the-mill desktop computer running text searching software.

Electronic publishing also has its down side, at least for the time-being. Since ejournals are the excep-

tion rather than the rule, many of the advantages listed above have not yet been realized by scholars in most disciplines. Moreover, coverage of topics outside of science and technology is sparse, and it's difficult to know what's available unless you happen to trip upon one of the few catalogs of electronic journals (we'll trip upon them later). Then there is the issue of

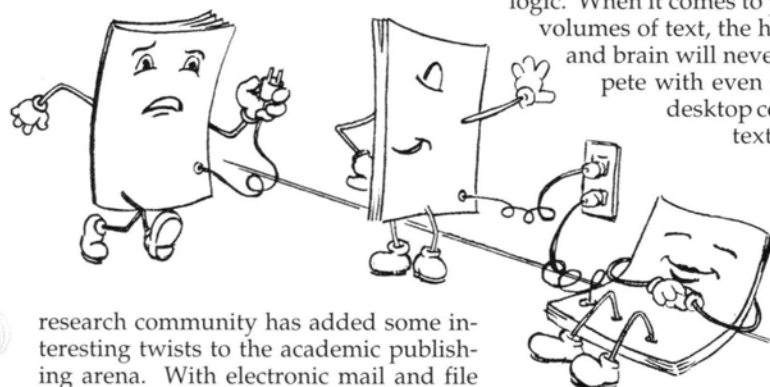
quality control. Virtually anyone with an e-mail account can publish on the Internet. Even if a title appears in an ejournal catalog, there is no guarantee that the publication is good or even mediocre. Only a handful of electronic journals are subject to any type of peer review, so in most cases there is no board of distinguished scholars making decisions about what does and doesn't get published. Finally, the visual presentation of many ejournals leaves something to be desired. Most are produced as plain ASCII text files and lack the tables, figures, illustrations, and typographical effects that bring the printed page to life. Although most of these problems will undoubtedly be solved within the next several years, for now we must either accept the drawbacks of ejournals or ignore a burgeoning outlet for serious scholarship.

### EJournal Catalogs

If you would like to know the names and types of electronic journals available today, there are several sources to which you can turn. Michael Strangelove at the University of Ottawa compiled and maintains the *Directory of Electronic Journals and Newsletters*. This document identifies, describes, and provides instructions on how to subscribe to roughly 140 different electronic publications on topics as diverse as drosophila, ethnomusicology, postmodern culture, librarianship, and so on. A lengthy introduction provides information on other network resources for electronic publications, a bibliography on the present and future ramifications of electronic publishing, and an informative article by Charles W. Bailey, Jr. entitled "Getting an ISSN [International Standard Serial Number] for an Electronic Journal." Strangelove's directory can be retrieved in two ways, by e-mail or anonymous ftp. To request the directory through e-mail, send a message to [listserv@uottawa.bitnet](mailto:listserv@uottawa.bitnet). The body of the message should consist of the following two lines and nothing else (note the spelling of **ejournal** and **directry**):

```
get ejournal1 directry
get ejournal2 directry
```

The directory will be sent to your e-mail address as two separate, rather lengthy messages. You can also grab the Strangelove directory from the anonymous ftp server [ftp.eff.org](ftp://ftp.eff.org). Change to the **/pub/journals** directory and get the file called **EJournal.Directory2.1.Z**. This file is in



research community has added some interesting twists to the academic publishing arena. With electronic mail and file transfer, now scholars can easily share data and collaborate on articles over long distances without the delays associated with sending disks and paper over land and sea. Computer networking has also given rise to a new and important publication medium, the *electronic journal*.

The term electronic journal, or *ejournal*, is used among librarians and bibliogra-

(continued on page 7)



## EJournals...

(continued from page 6)

UNIX compressed format and must be uncompressed before it can be used.

There are two other well-known directories that include descriptions and subscription information for electronic publications—the Kent State *Directory of Scholarly Electronic Conferences* (copyright 1992 by Diane Kovacs) and the so-called *List of Lists*. Unlike Strangelove's directory, which limits itself to serial journals and newsletters, these works also catalog moderated and unmoderated e-mail discussion groups. If you're just looking for ejournals, you'll have to sift through many unrelated items when working with these two documents, but both are invaluable tools for anyone looking for specialized information on the Internet.

The Kent State directory is on the anonymous ftp server **ksuvxa.kent.edu** in the **library** directory (**ksuvxa** is a VAX/VMS system, so, even though the name of the directory is **LIBRARY.DIR**, use the command **cd library** [not **cd library.dir**] to change to the directory). The catalog itself is broken up into eight files, **ACADLIST.FILE1** through **ACADLIST.FILE8**, each covering one or several major subject areas. To find out what is in each file, get and read the file called **ACADLIST.README**. This file will tell you how to retrieve the other files using ftp or e-mail. It also gives general instructions on how to subscribe to electronic publications and conferences.

The *List of Lists* is a directory of special interest group mailing lists available on the Internet and is updated quarterly. Entries are listed alphabetically by e-mail list name and provide more or less the same type of information as the Kent State directory. Because the list is not organized by subject matter, this resource is a little awkward to use, but a simple text searching tool like the **find** utility in your word processor or text browser can help you locate items of interest. An electronic copy of the *List of Lists* can be ftp'd from **ftp.nisc.sri.com**. The file, called **interest-groups**, is in the **/netinfo** directory and is roughly 1.2 MB. A hardcopy, indexed version of the catalog is published by Prentice Hall under the title *Internet Mailing Lists* (ISBN 0-13-327941-3). It can be ordered directly from the publisher by calling 1-515-284-6751.

## Subscribing to EJournals

Most electronic journals are distributed to subscribers by e-mail. Usually the person or group responsible for the publication makes arrangements to set up a mail reflector or **LISTSERV**, which makes it possible to use a single command to send

out an issue to all subscribers. Although the procedure for subscribing to particular journals may vary, the usual method is to send an e-mail message to a designated address (e-mail addresses are provided in the catalogs listed above). The body of the message should contain the text **subscribe listname Your Name** (e.g., **subscribe psyc Sigmund Freud**) or some variation of this basic command. If you normally append a signature to your e-mail messages, leave it off. Subscription requests are often handled automatically by list processors, which will interpret the lines of your signature as bad or unknown commands. After subscribing, you will receive e-mail notification that your name has been added to the list or that your subscription could not be processed. In the latter case, read the error message and try subscribing again, making the necessary adjustments.

## EJournal Archives

Back issues of ejournals are often archived and can be obtained by e-mail or ftp. There are also several sites on the Internet that hold extensive collections of electronic publications. The newest and most comprehensive collection is managed by Billy Barron, well known for his directory of online library catalogs (*UNT's Accessing Bibliographic Databases*). It resides on a host owned by CICNet, a regional network serving Big Ten universities and other institutions in the Midwest. The CICNet collection is accessible by Gopher (from the UIUC Gopher Main Menu, go to **Other Gopher and Information Servers/USA/All/CICNET gopher server/Electronic Serials**). This gopher menu offers access to over 200 serial publications. You can look for items alphabetically, by general subject heading, or by Library of Congress subject classification. If you want to collect all of the back issues of a single journal from the CICNet archive, it may be faster to ftp to **ftp.cic.net**. The journals are organized alphabetically in subdirectories beneath the **/pub/nircomm/gopher/e-serials/alphabetic** directory.

The bulk of the CICNet electronic serials collection currently derives from two other ftp archives, **red.css.itd.umich.edu** (maintained at the University of Michigan by Paul Southworth) and **ftp.eff.org**, a host maintained by the *Electronic Frontier Foundation*, an organization dedicated to issues of privacy, civil rights, and public policy as they pertain to national and international computer networks.

## How to Start Your Own EJournal

If you, as an individual or member of a professional society, are interested in publishing an electronic newsletter or journal, CCSO and other organizations can pro-

vide you with advice and the tools necessary to get started. Before you begin production, you may want to acquaint yourself with common electronic publishing conventions along with the problems typically associated with this medium. The e-mail discussion list **vpiej-l@vtvm1.bitnet** is for people specifically interested in scholarly electronic journals and electronic publishing. To subscribe, send an e-mail message to **listserv@vtvm1.bitnet** and enter the following text in the body of your message: **sub vpiej-l Your Name**. Another useful resource is the peer-reviewed electronic journal called *Ejournal*, which, according to the Strangelove directory, focuses on "theory and praxis surrounding the creation, transmission, storage, interpretation, alteration and replication of electronic text." Back issues of *Ejournal* are available on the CICNet gopher server. To subscribe to *Ejournal*, send e-mail to **listserv@albnyvm1.bitnet** and enter the line **sub ejrnl Your Name** in the body of the message.

CCSO staff can provide you with information on how to convert word processing files to plain ASCII text for dissemination on the net and how to format your publication for optimal presentation. For assistance along these lines, contact Lynn Ward at **L-ward1@uiuc.edu** or Lynn Bilger at **c-bilger@uiuc.edu**.

Once you are ready to begin distribution, you should establish an electronic mailing list on CCSO's main list server. Non-commercial, university-related mailing lists can be set up for faculty and staff at no cost. The list manager software (a UNIX program called *MajorDomo*) will automatically add new subscribers to your mailing list or, in the case of private lists, send the subscription request to you (the list owner) for approval. When ready to send out an issue of your publication, simply create an e-mail message addressed to your list name (e.g., **esoteric\_news@listserv.cso.uiuc.edu**) and incorporate the text of your ejournal into the body of the message. All list subscribers will receive a copy. For more information on setting up and managing a mailing list, contact Paul Pomes at **p-pomes@uiuc.edu**.

Electronic publishing is just one of many ways to make information available on the Internet. Unlike other types of Internet information services, managing a mailing list requires very little technical knowledge and is an efficient and inexpensive vehicle for disseminating information worldwide. ■

- Lynn Ward

# Once Upon A Packet Driver...

## ADMINISTRIVIA



Packet drivers are a daily part of life for the modern network administrator. They're just another one of those voodoo magic things that net admins configure, and, when the gods are pleased, it all works. We, however, have heard the story of the packet driver and know what it is supposed to do and why. Lean closer to the page (or screen depending on how you're reading this), and we'll tell you the tale of the wily packet driver.

Once upon a time, back in the olden days, when LANs were everything a lab of IBM PCs and clones could hope to become, users only ran one network program at a time. These were not modern times like today, when you might be logged into a few servers and have telnet and PC Gopher sessions running, all while trying to ftp a large file from a site in Austria. No, these were simpler times when your **autoexec.bat** file consisted of a PATH statement and an additional line or two that logged you into your LAN server.

One day, the Internet became available to these isolated LANs. Entire buildings were networked so that individual PCs could access Internet hosts across campus and around the world. At around the same time, a strange and new program was distributed—telnet it was called. A few brave network administrators installed telnet on their lab machines, configured it according to the instructions, tried to connect to a campus mainframe to read their e-mail and... their PCs froze up impressively. It seemed that the network card was unable to run two network programs simultaneously, so the programs fought it out to determine who (if either) would ultimately have access to the network. For a while, it was impossible for applications using the TCP/IP protocols (such as telnet and ftp) and programs using IPX (Novell NetWare's native transport protocol) to coexist, but eventually a software utility called the packet driver came along to keep the peace.

The packet driver promised to mediate access to the network interface card so that all network programs could get along. Of course, network applications had to promise to recognize the packet driver and abide by its wishes. Telnet and IPX (as well as other network programs) were rewritten to check for packet drivers and to give the packet driver control of the network card.

The packet driver, in turn, dutifully directed IP packets to the TCP/IP application and server related packets to the server program. So today, if a packet driver is installed, many network applications can coexist happily ever after.

## Stalking the Wily Packet Driver

**Vocabulary Alert.** Packet drivers circulate under several assumed names, so you may not recognize a collection of them, even when they are right under your nose. This is because many people and organizations have been involved in the evolution of the packet driver, and their names are often prefixed to the collection name as a whole. The original specification was created by John Romkey and is currently maintained by James van Bokkelaan at FTP Software, Inc. The collection of drivers themselves was originally supported and maintained by Russ Nelson, who previously worked at Clarkson University and now works at Crynwr Software. A packet-driver compatible version of IPX was developed at Brigham Young University. Thus, packet drivers are often referred to as the "FTP Software packet drivers," the "Clarkson packet drivers," the "Crynwr packet drivers," and even the "BYU packet drivers." The current term for the large public domain collection of packet drivers is the Crynwr collection, but some ftp sites still call them the Clarkson drivers.

In addition to the generic name for the collection, there are many different flavors of packet drivers, at least one if not several for every brand and model of Ethernet card, and each has a different filename. So, where do you find them? Some Ethernet card vendors provide a packet driver along with the software that comes with the card. For example, Standard Microsystems Corporation (the company that took over Western Digital's Ethernet card division) includes the **8003PKDR.EXE** driver on their diagnostics diskette.

The largest single collection of public domain packet drivers is the Crynwr collection. The Crynwr drivers can be found on hundreds of anonymous ftp hosts on the Internet. A reliable source for the most recent drivers is **wuarchive.wustl.edu**, which is a mirror of the well-known SIMTEL20 archive. The collection is found in the **mirrors/msdos/pktdrvr** directory. The file **00-index.txt** in this directory lists all of the files in the directory and their purpose (this file is reprinted in Table 1 on page 9). The archive named **drivers.zip** contains the executable files and detailed

documentation for the Crynwr collection.

If for some reason **wuarchive.wustl.edu** is unreachable, another good anonymous ftp source for packet drivers is **vax.ftp.com**. Public domain drivers are in the **/pub/packet.driver/pubdom** directory and Original Equipment Manufacturer (OEM) drivers are in the **/pub/packet.driver/oem** directory.

## Configuring the Wily Packet Driver

The packet driver has to know the exact details of the Ethernet card with which it is going to communicate in order to correctly pass commands and packets to the right places. The typical syntax for loading a Crynwr packet driver is:

```
driver_name software_interrupt [hardware_interrupt]
[I/O_address] [RAM_address]
```

where *hardware\_interrupt*, *I/O\_address*, and *RAM\_address* are in hexadecimal notation and correspond exactly to the settings of the Ethernet card (these settings are usually controlled by jumpers and/or dip switches on the card, or set by software that came with the card). For example:

```
wd8003e 0x60 0x5 0x280 0xd000
```

The syntax for each different packet driver varies slightly, but it is always useful to know these parameters when installing a packet driver or any other software or hardware that uses memory and hardware resources (upper memory blocks, interrupt requests, etc.) on your PC.

If you are installing a brand new card, you must be sure that the hardware interrupt and memory settings do not conflict with any other devices in the machine. Many Ethernet cards, however, were installed in pre-packet driver days, which means you'll have to use some method of determining the settings of the card. This may involve running diagnostic software or opening up the machine and using the documentation for the card to check what the jumper settings mean. As always, write down the settings, if you haven't already, and put them in a safe place for later referral.

One tip, from us to you—remember that once you've loaded a packet driver, programs that try to access the Ethernet card (including the card's own diagnostic programs) usually bump into the packet driver instead of the card. This means that if you misconfigure your packet driver

(continued on page 9)

## Packet Drivers...

(continued from page 8)

and then try to use a diagnostic program to see what you did wrong, the diagnostic program will see the settings of the packet driver software, not the Ethernet card itself. Remember to comment out any packet driver calls in the **autoexec.bat** file before trying to figure out what went wrong.

### Nifty Tricks for You and Your Packet Driver

We've yet to touch on one critical piece of information that is usually given when the packet driver is loaded, the *software interrupt*. What is a software interrupt anyway? The short answer is, just set it to **0x60** and don't worry about it. The long answer is that packet driver-compatible programs look at a point in memory for sending and receiving commands. Normally, the location is **0x60** and many network programs check this location by default. The software interrupt is not something you set on the Ethernet card; it is purely software related, and really packet driver related at that.

#### Using Packet Drivers and Windows.

If you want to run telnet from within Windows, you'll have to understand a little bit more about the software interrupt. Among other things, Windows is a memory manager. Windows is able to use both extended and virtual memory to load very large programs and/or run multiple DOS sessions. This involves moving data in and out of the conventional 0-640 KB range so DOS programs can get to data as needed. Unfortunately, it also shuffles around the area of memory that the packet driver resides in and strangles it. What's needed is a program to communicate between Windows and the packet driver, a Windows-packet-driver driver. Happily, there is such a beast named **WINPKT.COM**. It comes with the Crynwr collection and is also commonly distributed with some network programs for the PC.

Remember how we said you could set your packet driver software interrupt to **0x60** and not worry about it? If you are running Windows, forget we even mentioned it. Telnet, IPX, and Trumpet, etc. all look for the packet driver at the software interrupt **0x60** by default. Normally, you'd put the packet driver there, but your machine will lock up if you install the packet driver without WINPKT. So, you put the driver somewhere else (say, **0x62**), and put WINPKT at **0x60**. WINPKT forwards all of telnet's inquiries to the packet driver which in turn keeps track of exactly which network protocol (i.e., TCP/IP, IPX, XNS, etc.) is used by which packet. So, if you're going to run packet driver-aware applications in Windows, you should load a packet

driver and WINPKT in your **autoexec** file. The two lines might look something like this:

```
wd8003e 0x62 0x5 0x280 0xd000
winpkt 0x60 0x62
```

Confusing? Yes, but not nearly as confusing as trying to run more than one TCP/IP program at a time with packet drivers.

**Running Multiple TCP/IP Applications under Windows.** One of the big advantages of the Windows environment is that you can run multiple applications simultaneously. Suppose that, while you are sorting a large database in one window and downloading a large file with ftp in another, you'd like to send a file to a printer on your Novell network and read your e-mail on ux1. The database is not a network operation, so it can go about its business without worrying about the other three applications. The packet driver will let you print while accessing your e-mail, and WINPKT lets you do both operations while within Windows. But what about the ftp session? Well, packet drivers can distinguish between different network protocols like IPX and IP, but not the programs that use them. So, if you try to run more than one TCP/IP program (e.g., telnet, ftp, NUPOP, PC Eudora, etc.) simultaneously in Windows, the packet driver will assume all IP packets belong to just one of those applications. When this happens, either none of the programs will work or your machine will lock up completely. To solve this problem, Graham Robinson developed a *packet multiplexor* program called PKTMUX.

PKTMUX recognizes multiple network programs by assigning a separate channel

number to each network application in use. Like WINPKT, PKTMUX directs the correct packets to the correct applications, but it has the additional benefit of being able to differentiate between multiple programs using the same network protocol. With PKTMUX loaded, you can run up to eight TCP/IP programs at once (or a combination of TCP/IP programs and applications using other protocols) and each will be assigned a unique channel number.

Installing PKTMUX is not for the faint of heart. Configuration depends on exactly which network applications you plan to run and many other variables. The software comes with a 50 page document providing basic installation instructions and examples of how to use PKTMUX with various application combinations. Software and docs are bundled together in the file called **pktmuxx.zip** (where x is the current version number) and is available at **wuarchive.wustl.edu** as shown in Table 1.

### Sigh!

So, let's review. In order to run most public domain TCP/IP packages or to run a TCP/IP program together with another network protocol, you need a properly configured packet driver, and all of your network applications must be packet driver-aware. If you want to run packet driver-aware applications under Windows, you need to know the software interrupt assigned to your packet driver and need to install WINPKT. Finally, if you want to run several TCP/IP programs concurrently, you need to install PKTMUX. Confusing? Tricky? Well, yes. Kind of makes you long for the olden days. ■

-Dave Ruby and Lynn Ward

**Table 1**  
**00-index.txt**

Directory PD1:<MSDOS.PKTDVR>					
Filename	Type	Length	Date	Description	
3c509a.zip	B	53583	930216	3c509(EtherlinkII)packetdriver,v1.0.1	
dis_pkt9.zip	B	32264	921211	PacketdriverthatrunsoveranNDISdriver	
drivers.zip	B	338294	920201	Crynwr10.0packetdriverexecutable&docs	
drivers1.zip	B	268293	920201	Crynwr10.0packetdriversw/sources,part1	
drivers2.zip	B	149548	920201	Crynwr10.0packetdriversw/sources,part2	
drivers3.zip	B	61432	920315	Crynwr10.0packetdriversw/sources,part3	
ipxtpip.zip	B	26703	920618	ConfigurationhelpforNovellIPXPKTusers	
novell.zip	B	26173	920402	BYU'spacketdriverIPX(triedandtrue)	
pdclk145.zip	B	99509	921007	SetsPCclockusingUDP/IPtimeserverviaLAN	
pdether.zip	B	29000	920402	Intel'spacketdriverIPX	
pdipx103.zip	B	15323	920929	PacketdrivershellbyIntelforNetWare	
pktmux12.zip	B	72021	930215	RunsimultaneousTCP/IPsessionswithDV/Win3x	
popml272.zip	B	200991	920623	POPmail:TCP/IPemail,useCrynwrpktdrivers	
slippr13.zip	B	14243	930228	TCP/IPpacketdriversforserialSLIP&CSLIP	
telnetd.zip	B	29365	920617	RemotetelnetlogintoPCconsoleviaTCP/IP	
trmp105g.zip	B	131992	920807	NNTPUsenetnewsreader.UsesCrynwrdrivers	



# FTP and VMD

## NET TIPS



(Editor's Note: This article assumes a working knowledge of ftp. If you are not yet familiar with the ins and outs [or puts and gets] of ftp, see Appendix B of CCSO User Guide #401, FREE SOFTWARE•How to find it•How to get it•How to make it usable (uncompress it), available at the CCSO Resource Center, 1420 DCL and at most CCSO sites.)

Mastering ftp, the TCP/IP file transfer utility, is no small task. Aside from memorizing a number of obscure commands, in order to use ftp successfully, you also need to know about different operating systems, file systems, file naming conventions, file compression formats, and lots of other technical minutia. Using ftp to transfer files to or from VMD (or any IBM mainframe running the VM/CMS operating system) presents some additional obstacles. In the following paragraphs, we'll explore some of the common questions and problems that arise when trying to use ftp with VMD.

### Client versus Server

Regardless of platform, one of the most difficult ftp concepts to grasp is which end of the connection is the client (or local host) and which the server (or remote host). The point of confusion arises because your desktop computer can be the client, the server, or neither, depending on how you start the transfer. A good rule of thumb is that the client is always the computer from which you initiate the transfer. For example, if you open an ftp session with VMD by typing a command like `ftp vmd.cso.uiuc.edu`, the computer on which you executed the command is the client. If you type the command from the operating system prompt of your desktop PC or workstation, your desktop computer is the client and VMD is the remote host. Similarly, if you start an ftp application that asks you to enter the name of the host you want to contact (such as *Fetch* for the Macintosh or *WFTP* for Windows), the client is the machine on which that application is installed, usually your desktop workstation.

Things get a little more complicated when you start an ftp session from within a telnet or tn3270 session. Under these circumstances, the computer into which you are logged is the client. For example,

let's assume you've used a tn3270 program to log in to VMD. If you execute the `ftp` command at the VMD system prompt in order to open an ftp session with `ux1`, VMD is the client and `ux1` is the server. On the other hand, if you had used telnet to log in to `ux1` and started an ftp session with VMD from the UNIX system prompt, `ux1` is the client and VMD is the server. Note that in both of these cases, your desktop computer is neither client nor server.

Users often want to move files from VMD to their desktop computer. This is where things get a little hairy. If you've logged in to VMD with tn3270 software from your desktop computer and then execute an `ftp` command at the VMD system prompt (or press a special key combination) to open an ftp session with your Mac or PC, VMD is the local host and your desktop computer is the remote host or server. In a geographical sense, this is counter-intuitive because your PC is "more local" than VMD. But during an ftp session within a tn3270 session, your PC acts as if it were two different pieces of equipment: 1) a terminal connected directly to VMD and 2) a host on the Internet running an ftp server. Whenever you ftp from VMD to another machine, your terminal tells VMD to open an ftp session with another node on the Internet. It just happens that, in this case, the host running the ftp server is your desktop computer, but it could just as easily be a mainframe in New Zealand. VMD doesn't really know or care where the ftp server is located; it only knows its name or IP address.

When VMD is the client, the `put` command is used to move files from VMD to the server and the `get` command is used to move files from the server to VMD. When VMD is the server, the reverse is true—that is, the `put` command is used to move files from the client machine to VMD and the `get` command is used to move files from VMD to the client. Either way, there are some special issues to consider when moving files to and from VMD.

### Filenames

Every operating system has its own rules for naming files. Filenames on VMD (and other VM/CMS systems) consist of three parts separated by spaces, for example, `UNREAD NOTEBOOK A0`. The first part may contain no more than eight characters and is the **filename** proper. The second part describes the contents or nature of the file and is thus referred to as the **filetype**. The filetype is also limited to eight characters. The third part, the

**filemode**, uses one or two characters to indicate the minidisk on which the file resides and the status of the file. When naming files, both the first and second parts must be typed explicitly, but the **filemode** defaults to "A" (your personal minidisk) unless otherwise specified.

VMD filenames require special treatment during an ftp session. First of all, when typing a VM/CMS filename after a `put` or `get` command, the parts of the filename should generally be separated by a period rather than a space. Secondly, when moving a file "from" another operating system "to" VMD, you must be sure to give a valid target filename—that is, one that contains both a filename and filetype (and optionally a filemode). For example, if you are trying to `get` a file from a UNIX system called `README`, you would have to give both the original name and a VM/CMS-compatible target filename after the `get` command (e.g., `get README README.TEXT`). If the original filename has two parts already (e.g., `uiucnet.index`), you need not specify a target filename unless you want the file to have a different filename on VMD (e.g., `get uiucnet.index` or `get uiucnet.index uiucnet.text`). If the original filename has three or more parts separated by periods, the target filename must be changed so that it only has two parts (e.g., `get virus-detective.506-strings.hqx virus.detectiv`). Finally, if the first or second part of the original filename is longer than 8 characters, the extra characters will automatically be removed from the name when the file is transferred to VMD.

When transferring a file from VMD to another machine, the same rules apply. For example, to `put` the file called `UNREAD NOTEBOOK A0` onto another system, separate the parts of the name by a period rather than a space and leave off the filemode unless it is required because the file is on a minidisk other than "A." If you do not specify a target filename, the VM/CMS filename will usually be adapted to conform to the naming conventions of the remote system. For example, if you try to `put` the file `UNREAD NOTEBOOK A0` onto your PC, the resulting DOS filename will be `unread.not`.

### Transfer Mode: ASCII vs. Binary

Another peculiarity of VMD (and many other IBM systems) is that plain text files conform to the EBCDIC rather than the ASCII standard for using numerical val-

(continued on page 11)

## FTP and VMD

(continued from page 10)

ues to represent printable characters. In EBCDIC, an uppercase "A" has the decimal value of 193, while in ASCII the same character has the value of 65. In order to safely transfer a text file from an ASCII system to an EBCDIC system or vice versa, the ftp transfer mode must be ASCII. If the transfer mode is set to ASCII, the file will be converted from EBCDIC to ASCII or ASCII to EBCDIC on the fly. But if you set the type to binary, the resulting file will be indecipherable.

Sometimes it's obvious that a file is a plain text file. Files with names like **readme** or **index** or extensions like **doc**, **txt**, etc., are usually text files. Also, any readable file on your VMD minidisk is a text file. This includes your main configuration file (**PROFILE EXEC**), mail files, names files, and so on. There some additional file types that should also be transferred in ASCII mode. PostScript files, which often end with a **ps** extension, are text files. Two other common formats are BinHexed files (which often end with an **hqx** extension) and uuencoded files (which often end with **uue**). The latter two types are binary files that have been encoded into ASCII for safe transport over e-mail and bulletin board systems. These files too should be transferred in ASCII mode.

Files that have been archived or compressed with file compression software should usually be transferred in binary mode. Common file extensions for archived files are **Z**, **tar.Z**, **zip**, **sea**, **sit**, **cpt**, **exe**, **pak**, **zoo**, **lzh**, **lzw**, **lzs**, and **arc**.

## VMD as FTP Client

Many users find it easier to invoke the ftp client on VMD and ftp "from" VMD "to" other machines rather than the reverse. No matter where you're going, there is one method of starting an ftp session on VMD that will always work. First, link to the minidisk on which VMD's TCP/IP applications are stored by typing the command **linkto tcpip** at the system prompt. Then enter the command **ftp host**, where **host** is the IP address or fully-qualified domain name of the computer with which you want to communicate. Once connected to the host, you'll be prompted to enter your username (or "anonymous," for anonymous ftp archives) and perhaps your password. Once logged in, you will be presented with a **Command:** prompt and can enter any valid ftp command supported by the VMD client, including common ftp commands such as **dir**, **put**, **get**, **mget**, **ascii**, **binary**, etc. For a complete list of commands available to you, enter the command **help** at the ftp **Command:** prompt.

## Shortcuts

**DOS Users.** If you want to establish an ftp connection with your desktop computer, there may be a few short cuts you can take. For example, if you're using Clarkson University's **tn3270** program to conduct a remote login session with VMD, once you've executed the **linkto tcpip** command, you can initiate an ftp session by pressing **<Alt - t>**. This key combination will pass a string to the VMD ftp client that will automatically open a session with your desktop computer—in other words, you don't have to type the command **ftp host** at the system prompt. Then, if the ftp server on your PC is password protected, you can press **<Alt - w>** to automatically send an internal password from VMD to your server.

**Macintosh Users.** Although the standard ftp commands will work with the Mac, there is a set of utilities installed on VMD that make it much easier to transfer files to and from your Mac in conjunction with Brown University's **TN3270** software. Instead of linking to the standard TCP/IP minidisk, enter the command **linkto tcpip mac** at the system prompt. Then, to download a file from VMD to your Mac, enter the following command at the VMD system prompt:

```
wmac filename filetype [filemode]
[options]
```

where **filename** and **filetype** are the exact name and type of the file as displayed by the **filelist** or **listfile** commands on VMD (and **filemode** and **options** are optional). Note that the name, type, mode, and options are separated with spaces rather than periods; for example, **wmac all notebook**.

Once you've executed the **wmac** command, the standard Macintosh **Save** dialog box will appear. You can either accept the default filename and folder or change them as you would when saving any file to your Mac.

To move files in the other direction (from your Mac to VMD), use the command:

```
rmac filename filetype [filemode]
[options]
```

where **filename** and **filetype** are the name and type you want the file to have on VMD. A standard Mac **Open** dialog box will appear and you can select the file you want to transfer.

Files transferred using **wmac** and **rmac** are treated as ASCII files by default unless you add the **(binary)** option to the command; for example, **wmac pseudora sea (binary)**. To see a complete list of options for **wmac** or **rmac**, enter the command

**help cms wmac** or **help cms rmac** at the system prompt.

## VMD as FTP Server

The most complicated aspect of using an ftp client to open a session with the ftp server on VMD is the login procedure. Aside from knowing your login name and password, you must also know the name of your personal minidisk and your minidisk **write password**. The name of your minidisk is generally your **USERID** followed by **.191**. For example, the name of my minidisk is **lynnward.191**.

Your minidisk **write password** is something you must set up during a regular VMD terminal session with the **VMSECURE** utility. To establish minidisk passwords, follow these steps:

1. Log in to VMD with your terminal emulation software, usually some form of **tn3270**.
2. Link to the **VMSECURE** minidisk by entering the command: **linkto vmsecure**
3. Enter the command **vmsecure user** at the system prompt. You will be asked to enter your login password and then you will see a menu similar to that shown in Figure 1 below.

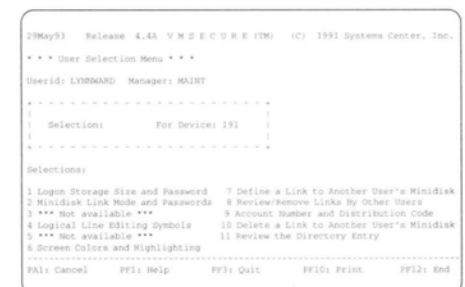


Figure 1: The VMSECURE User Selection Menu.

4. Select option 2, **Minidisk Link Mode and Passwords**. You will then see a screen similar to that shown in Figure 2.

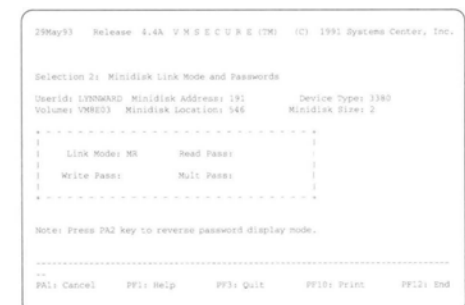


Figure 2: The VMSECURE Minidisk Link Mode and Passwords menu.

(continued on page 12)

RENEE ELLIS  
SPECIAL COLLECTIONS  
314 MAIN LIBRARY  
CAMPUS  
MC-522

## FTP and VMD

(continued from page 11)

5. Using the **tab** key, move to the **Read Pass:** field. Type a read password of eight characters or less and then **tab** to the **Write Pass:** field. Type another password of eight characters or less in this field. (It's a good idea to enter different passwords for read and write access to your minidisk.) Do NOT create a password for the **Mult:** field!
6. To view your newly created passwords, press your keyboard equivalent of the 3270 **PA2** key. If you are satisfied with them, press **F12** (or your keyboard equivalent) to save the passwords.

**Note:** Any VMD user who knows your minidisk passwords can link to your minidisk during his or her terminal session. Keep this information confidential unless you want others to be able to access your files.

Once you know your minidisk name and write password, you can open an ftp session with VMD using virtually any ftp client. The login procedure is described below (a typical login session is shown in Figure 3):

```
uxl 3: ftp vmd.cso.uiuc.edu
Connected to vmd.cso.uiuc.edu.
220-PTP2SERV2 at vmd.cso.uiuc.edu, 13:03:29 CDT SATURDAY 05/29/93
220 Connection will close if idle for more than 5 minutes.
Name (vmd.cso.uiuc.edu:ward): lynnward
331 Send password please.
Password: [password for lynnward is entered here]
230 LYNNWARD logged in; no working directory defined
Remote system type is VMD.
ftp> cd lynnward.191
550 Permission denied to LINK to LYNNWARD 191; still no working directory
ftp> quote acct private
230 Working directory is LYNNWARD 191
ftp>
```

Figure 3: A typical login session with the VMD ftp server.

1. Use your client to initiate an ftp session with the remote host **vmd.cso.uiuc.edu**.
2. Once connected, you will be asked to enter your username and password as usual. You will then see a message that looks something like "230 LYNNWARD logged in; no working directory defined."
3. Link to your personal minidisk by entering the command **cd userid.191**, where *userid* is your VMD login name; for example, **cd lynnward.191**. You will then see an error message like "Permission denied to LINK to LYNNWARD 191; still no working directory." Ignore this message and go on to step 4.

**Note for Fetch Users:** To link to your minidisk, go to the Remote menu and choose the **Send FTP Command...** option. Then enter the command: **cwd userid.191**, where *userid* is your VMD login name. You will see both of the error messages previously described, and another dialog box will tell you that there is still no working directory. Click on **Okay** and go on to step 4.

4. Enter the command **quote acct writepass**, where *writepass* is your minidisk write password. You will finally be connected to your minidisk and can use your ftp client as usual to manipulate files on the client and server.

**Note for Fetch Users:** To enter your write password, go to the Remote menu and choose the **Send FTP Command...** option. Send the command **acct writepass**, where *writepass* is your minidisk write password.

## For More Information

When it comes to ftp, VMD is a strange sort of beast. Aside from the points discussed above, you may encounter additional VMD-specific quirks. If you have problems or questions about doing file transfers between VMD and another computer, contact the CCSO Systems Consultants at 333-6133. ■

- Lynn Ward

## About UIUCnet

*UIUCnet* provides timely information about the campus network. It is published by the Computing Services Office and is edited by Lynn Ward. Comments and suggestions for topics are welcome. Permission to reprint all or part of *UIUCnet* for non-profit purposes is granted, provided full acknowledgement of the source is given. Feel free to reach us via electronic mail ([uiucnet@uiuc.edu](mailto:uiucnet@uiuc.edu)), campus or U.S. mail (UIUCnet, Computing Services Office, 1120 Digital Computer Laboratory, MC 256, 1304 W. Springfield Ave., Urbana, IL 61801), or by phone at (217) 244-0681.

For a free subscription, just send us your name and (preferably campus) address; be sure to include your department and mail code if you send a campus address. *UIUCnet* is also available in electronic form. Recent issues are posted to the Netnews newsgroup **uiuc.pubs.uiucnet**. Back issues are available for download from the anonymous ftp host **ftp.cso.uiuc.edu** in the directory **doc/net/uiucnet** and from the UIUC Gopher server.